Design and Development of a **Blind Person Assistant**

TEAM 6
Andy Lau
Emily Lai
Jeremy Mark Tubongbanua
Natasha Naorem

Advisor: Dr. Khalid Hafeez

N

1

The Problem & Research

The Problem

SAFETY

Reliable object detection

EFFICIENCY

Reasonable travel time

ORIENTATION

Self direction to detect surroundings and decide where to go

N

SURVEY QUESTION:

How well do available navigation tools meet your needs?

N

SURVEY CONCLUSION:

Compared to outdoor solutions, there is a gap in the market for development of suitable indoor orientation/navigation systems.

N

PROBLEM STATEMENT:

Current indoor navigation systems do not focus on human-centered design for visually impaired users.

Sighted Guide

LIMITATION: Availability

N

Current Solutions for Independent Navigation

OBSTACLE AVOIDANCE

Traditional Aids

- White Cane
- Guide Dog

ORIENTATION & NAVIGATION

Mobile Applications

- Google Maps
- Lazarillo
- Voice Vista

Stakeholder Discussion: Low Vision Student on Campus

Limitation: Disruptive Navigation

Simultaneous use of a white cane and mobile phone

N

PROJECT OBJECTIVES

Improved Obstacle Avoidance

- Increase detection range
- Detect overhanging obstacles
- Support the white cane

Non-Disruptive Navigation

- Reduce reliance on simultaneous use of navigation tools
- Improve navigation efficiency

2

Engineering Design Process

Design Requirements

When establishing a design, we wanted to meet certain requirements:

Keep the White Cane

Blind individuals rely heavily on their white cane, we do not want to remove or replace this crucial component.

Do Not Rely on a GPS

GPS is not reliant enough indoors to be used to a certain extent.

Assist with Indoor Navigation in Real-Time

The system would need to use multiple sensors and have some way of guiding the user indoors. It would need to conduct edge computing for real-time assistance.

Hands-Free Experience

If we were to build an app, it can be a hassle to hold both a phone and a white cane. We want to offer a hands-free experience so that the user's free hand can be used for better things

Use Cases

Our three key features that we offer to a visually impaired user assisting with indoor navigation.

Obstacle Detection & Classification

Detect objects in real-time, and notify the user through TTS (text-to-speech). The user can opt in for active detection or detection as needed.

Object Tracking

Guide the user to an object that was detected and classified. The system will use sensors and actuators to play a game of “hot and cold” and guide the user to a desired object.

Text Scanning

Many signs and poster boards are not visually impaired accessible. A text scanner would help with narrating texts, like large signs, using AWS Textract and TTS (text-to-speech).

E

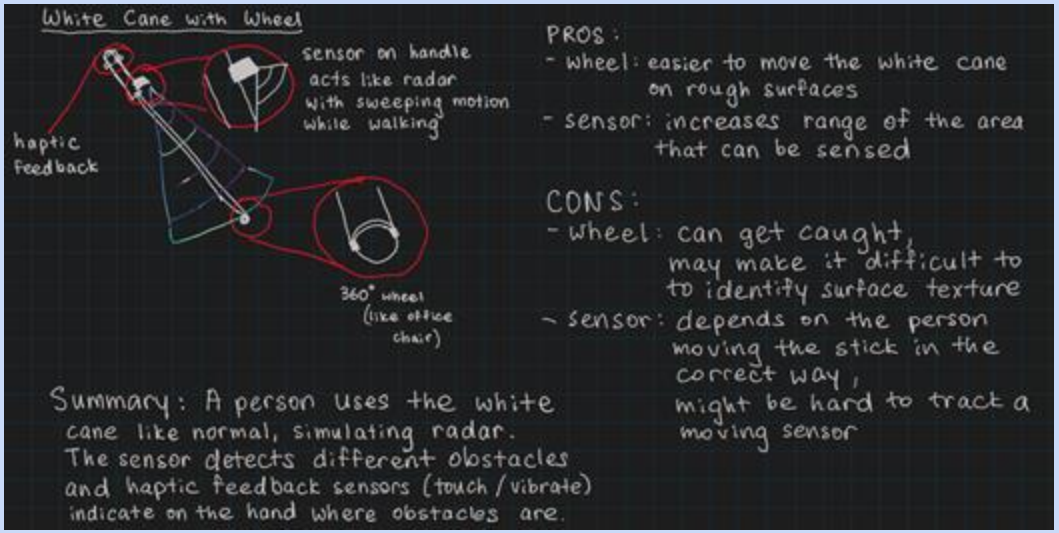
Design Iteration 1

Smart Cane with Omni Wheel at the Tip

- Wheel component for ease of cane movement
- Sensor increases range of area that can be sensed

Challenges

- Wheel changes the fundamental white cane design
- Moving sensor may be difficult to track



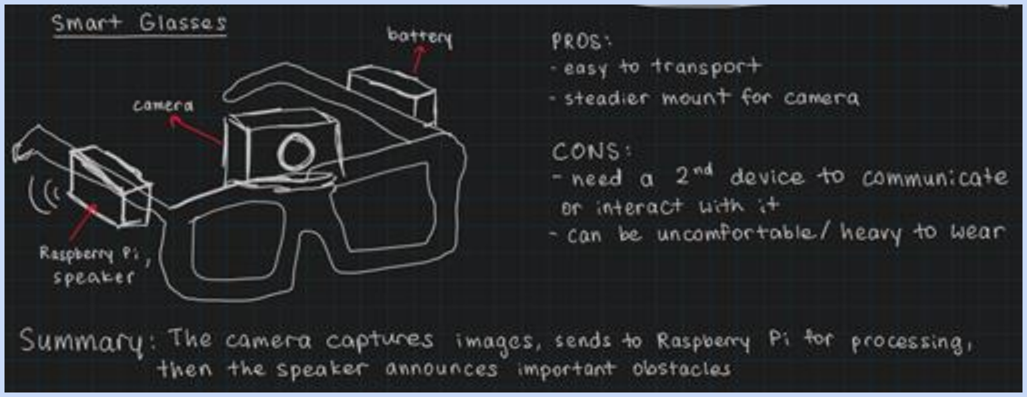
Design Iteration 2

Smart Glasses with Camera

- Camera captures images and sends to Raspberry Pi for processing
- Speaker announces important obstacles

Challenges

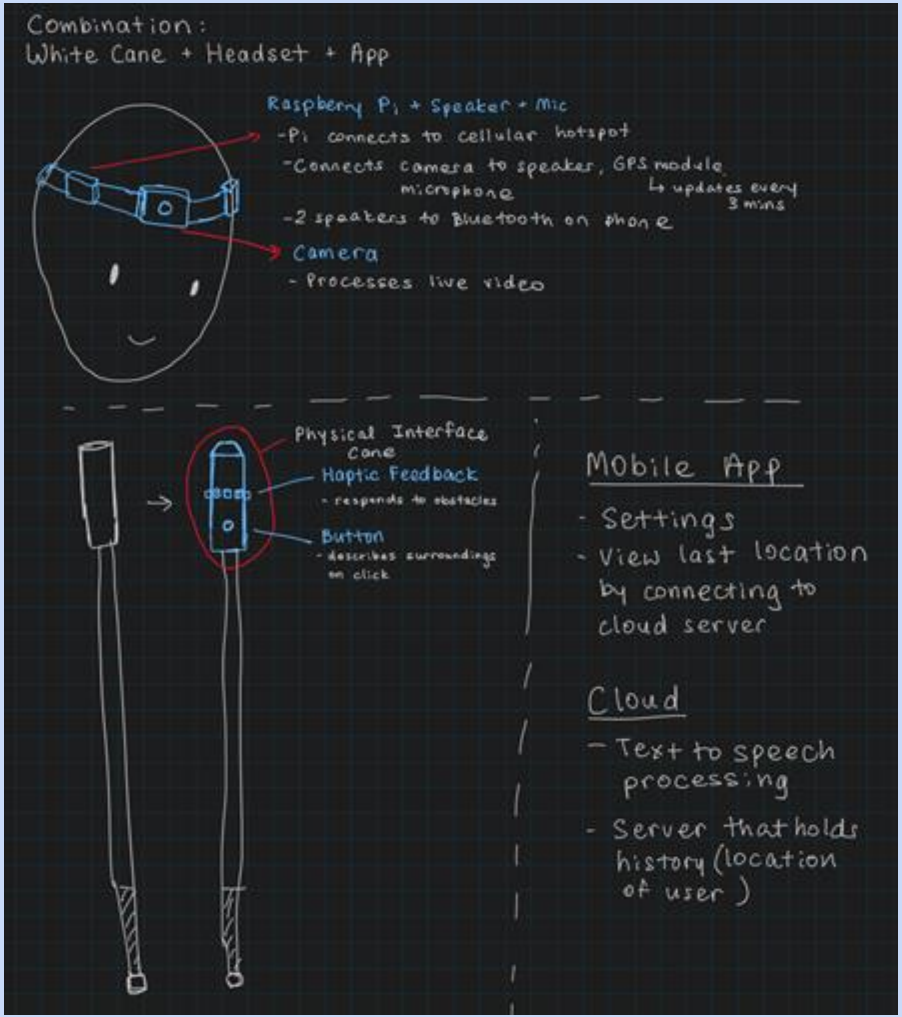
- Poor user experience due to imbalanced and heavy weight on the head



Design Iteration 3

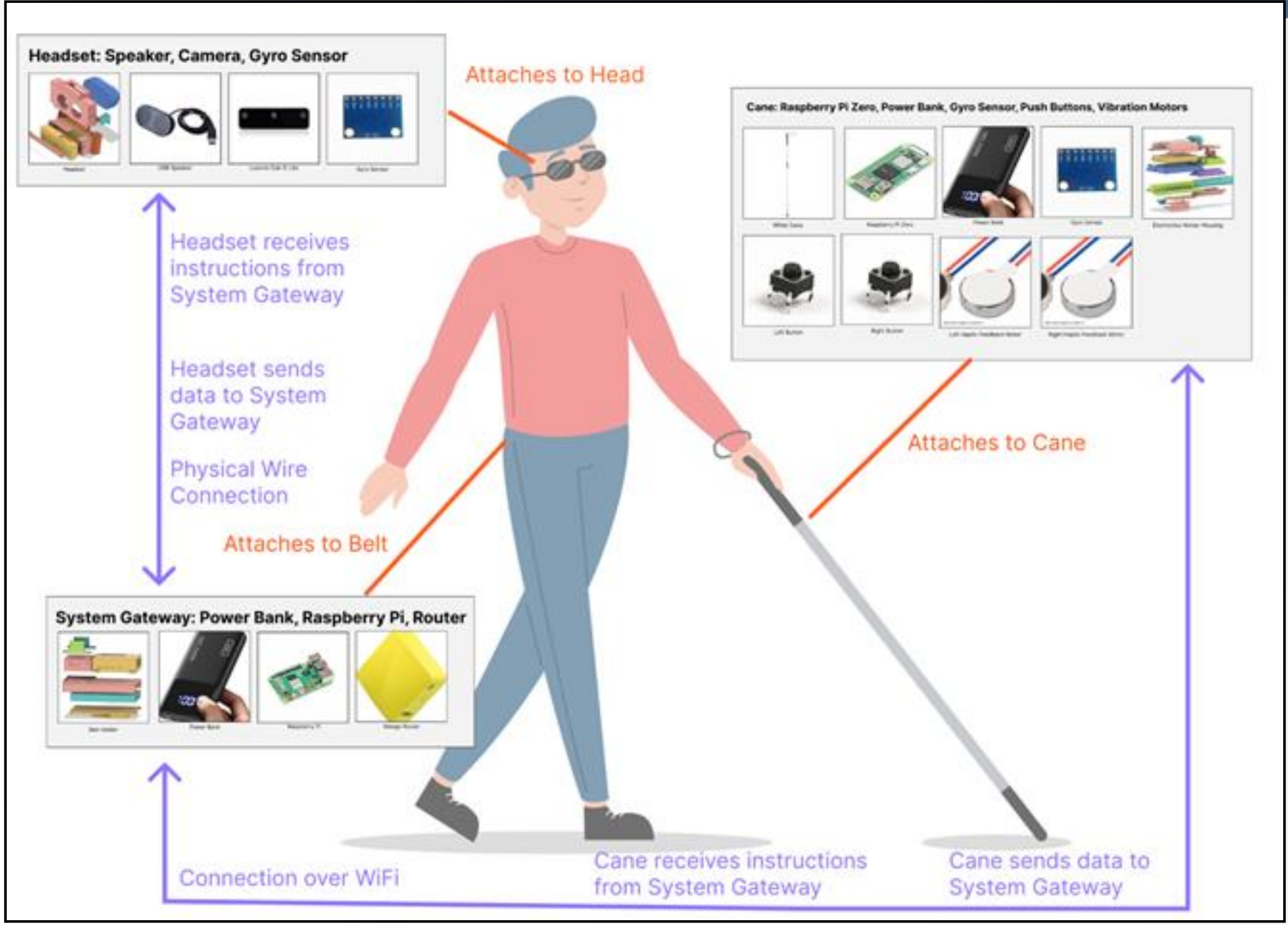
Smart Cane with Headset

- Headset
  - Raspberry Pi, speaker, microphone, camera
  - Can find obstacles and report audibly to the user
- Cane
  - Haptic feedback for obstacles
  - Button for user input
- Mobile App
  - For settings modification



The Design We Went With

- Headset
  - Camera
  - Speaker
  - Gyro Sensor
- System Gateway
  - Router
  - Raspberry Pi
  - Power Bank
- Smart Cane
  - Raspberry Pi
  - Vibration Motors
  - Push Buttons
  - Gyro Sensor
  - Power Bank



Luxonis Oak-D Lite

The core piece of hardware for our capstone.

⭐Has 3 Cameras:

1. Center 4K (12MP) RGB Camera
2. Left shutter stereo camera
3. Right shutter stereo camera

⭐Has an internal VPU (Visual Processing Unit). Edge Processing ✅

⭐Is Powered by USB-C and can transmit data through the same cable.

⭐Coding in Python and lots of documentation

E

Depth Perception

Using the Left and Right Stereo Cameras

Mimics human binocular vision to estimate depth

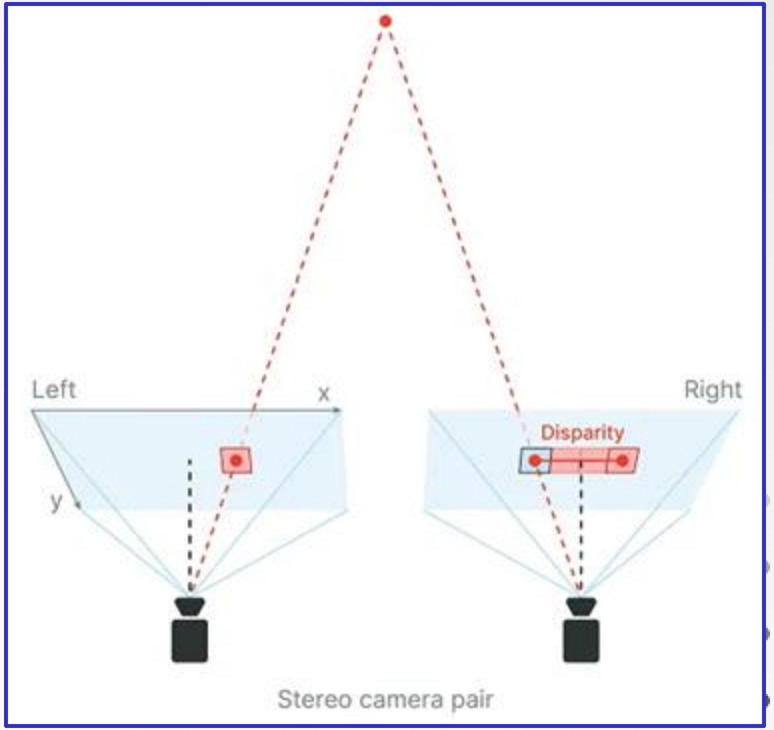
VPU generates a depth map

Oak-D lite comes with limitation perception range of [0.4m, 8m]

Examples for calculating the depth value, using the OAK-D (7.5cm baseline):

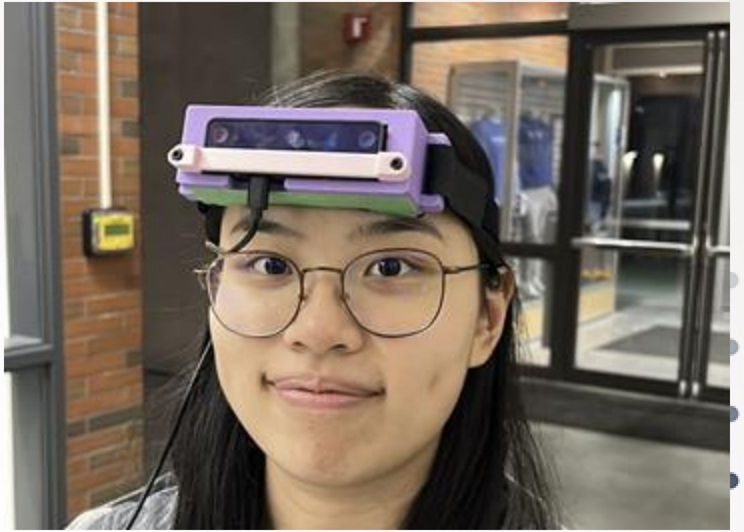
```Python
# For OAK-D @ 400P mono cameras and disparity of eg. 50 pixels
depth = 441.25 * 7.5 / 50 = 66.19 # cm

# For OAK-D @ 800P mono cameras and disparity of eg. 10 pixels
depth = 882.5 * 7.5 / 10 = 661.88 # cm
```

Note the value of disparity depth data is stored in `uint16`, where 0 is a special value, meaning that distance is unknown.



E

Headset Iteration 1



- Had a Horizontal Velcro Strap
- Had a Camera Enclosure & Lip

Improvements Needed:

- Needs to hold speaker
- Needs to hold a Gyro sensor
- USB cable is obstructive
- Needs a vertical strap hole for better stability on the head

J

Headset Iteration 1.5

Never became a full iteration, but we played around with different ideas.

We had trouble figuring out where to put the speaker.

Headset Iteration 2

New Features:

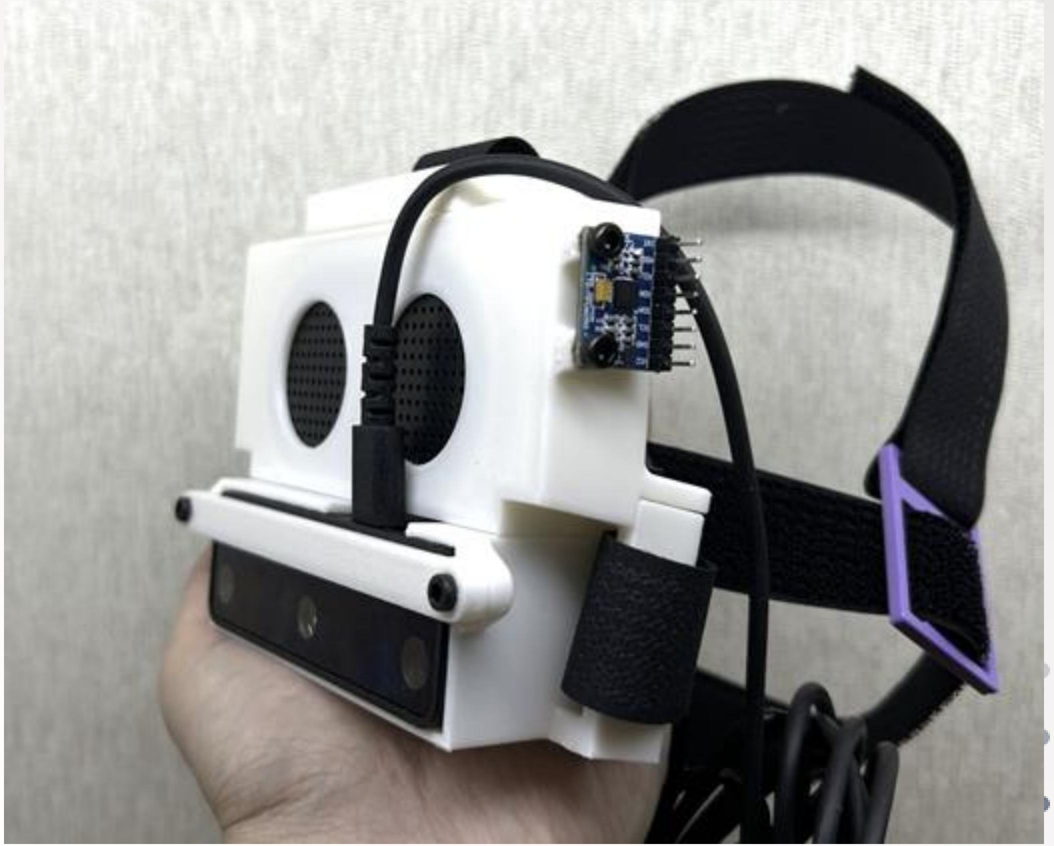
- New Gyro Seat
- New Snap-Fit Speaker Enclosure
- Camera was flipped upside down to help with wire routing (which originally blocked the user’s eyes)
- Has a vertical strap hole

J

Headset Iteration 3 (Final)

New Features:

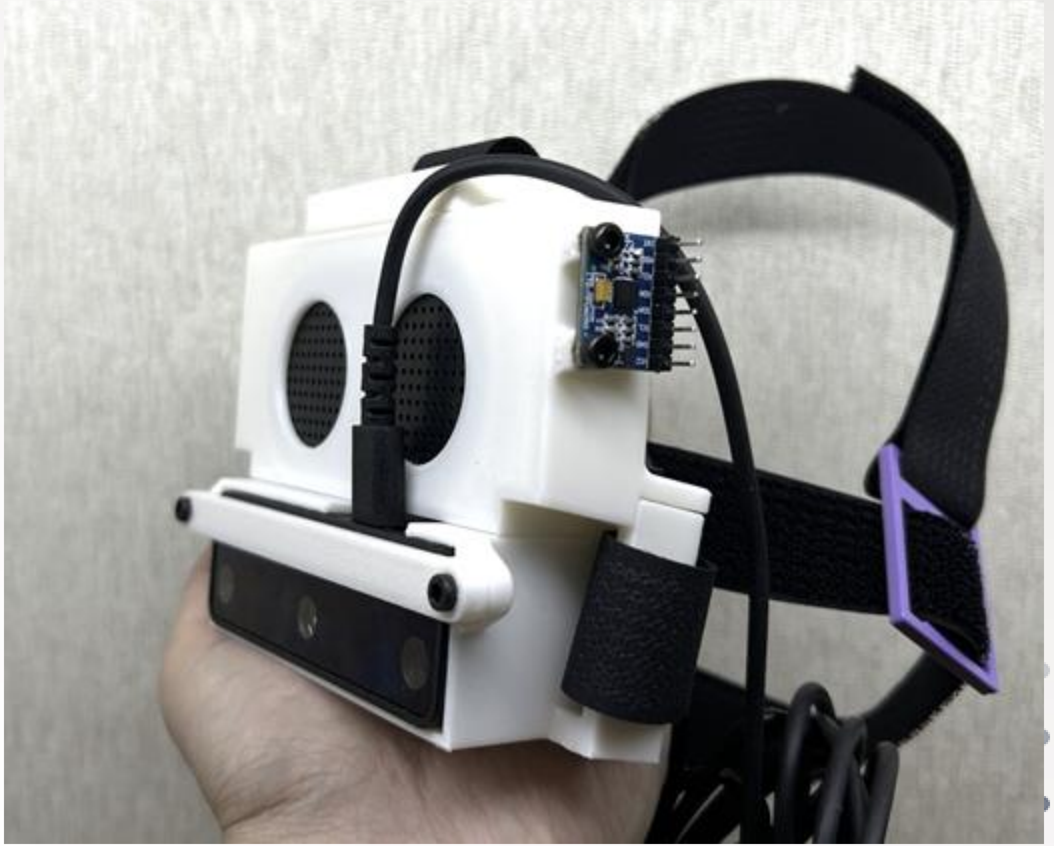
- Shifted Speaker Forward
- More Comfortable
- Angled Camera (more info in next slides)



Headset Physical Subcomponents

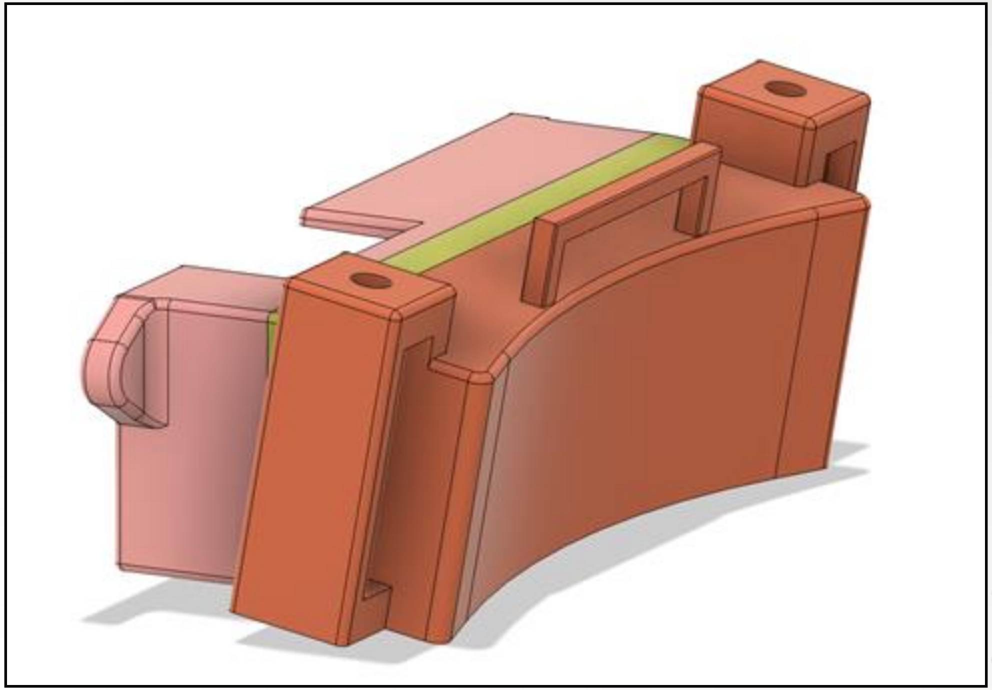
Put together using M3 heated inserts and M3 screws

1. Camera Frame
2. Camera Lip
3. Speaker Cage
4. Speaker Clamp
5. Gyro Seat



Subcomponent 1: Camera Frame

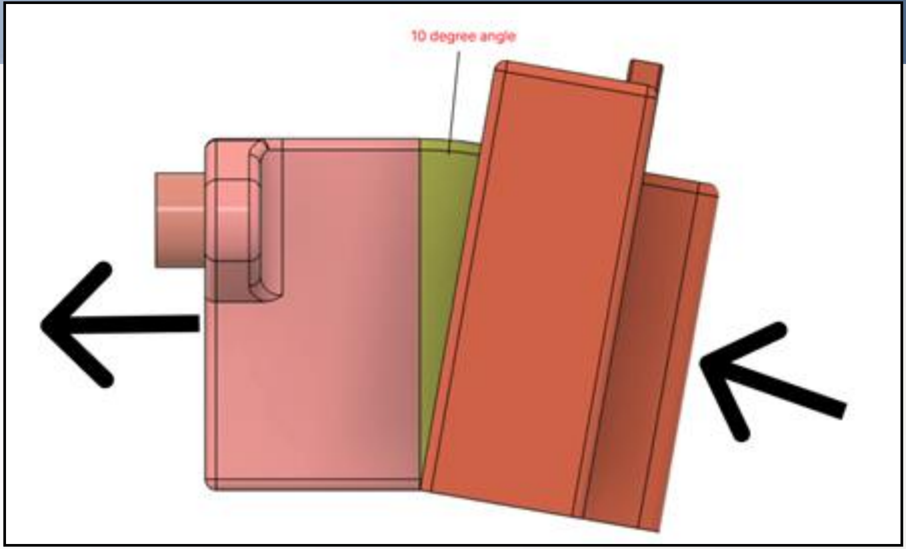
- Back of the enclosure is curved to fit most foreheads
- Has 2 rectangular holes for a velcro strap
- Slim-fit the Luxonis Oak-D Lite with 0.1 mm tolerance

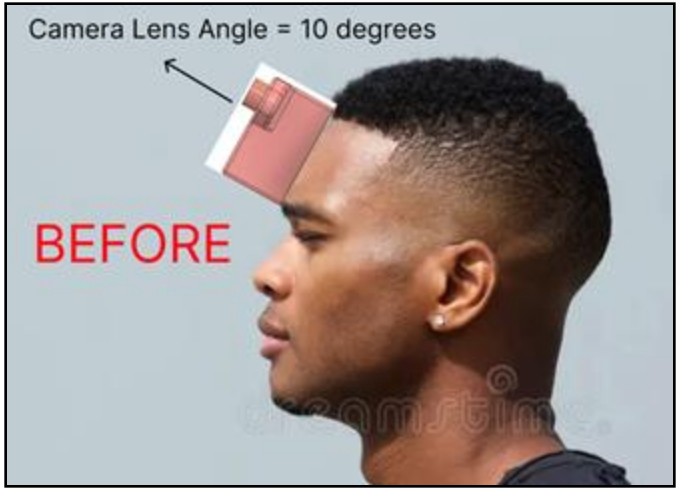


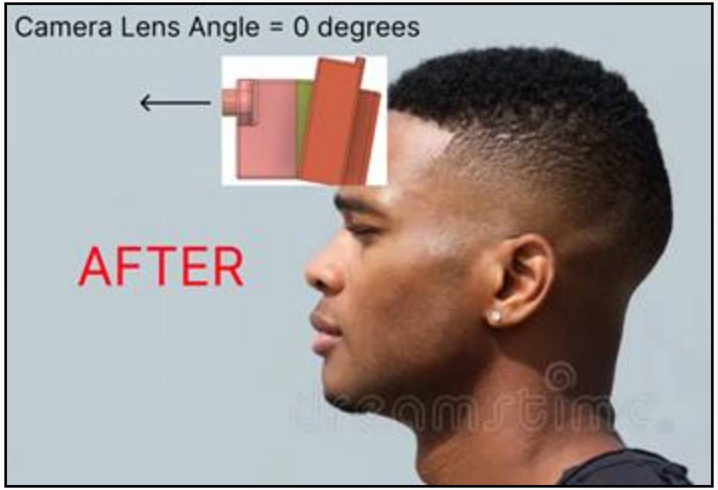
J

Subcomponent 1: Camera Frame

Having an angled camera frame optimizes the camera view.





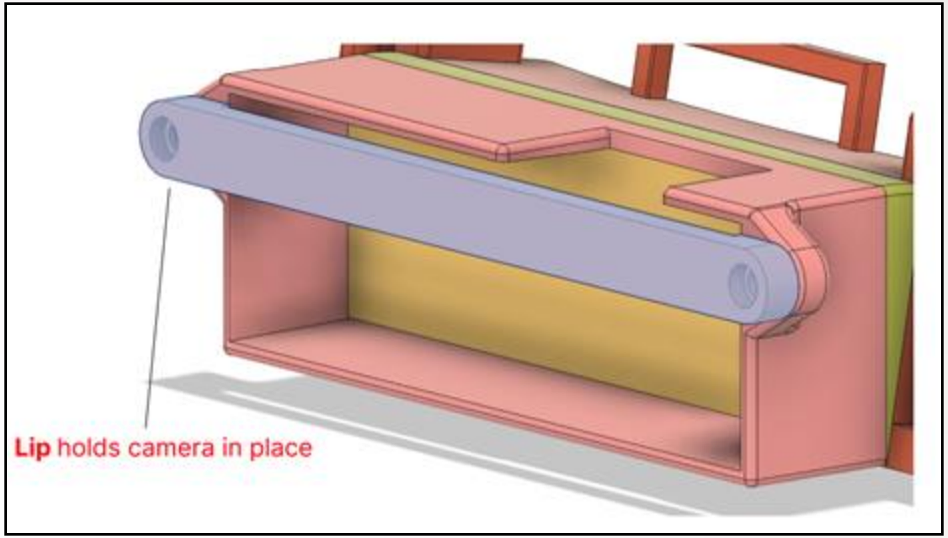


J

Subcomponent 2: Camera Lip

Lip holds the camera in place and prevents it from falling out of the enclosure, without blocking the 3 cameras.

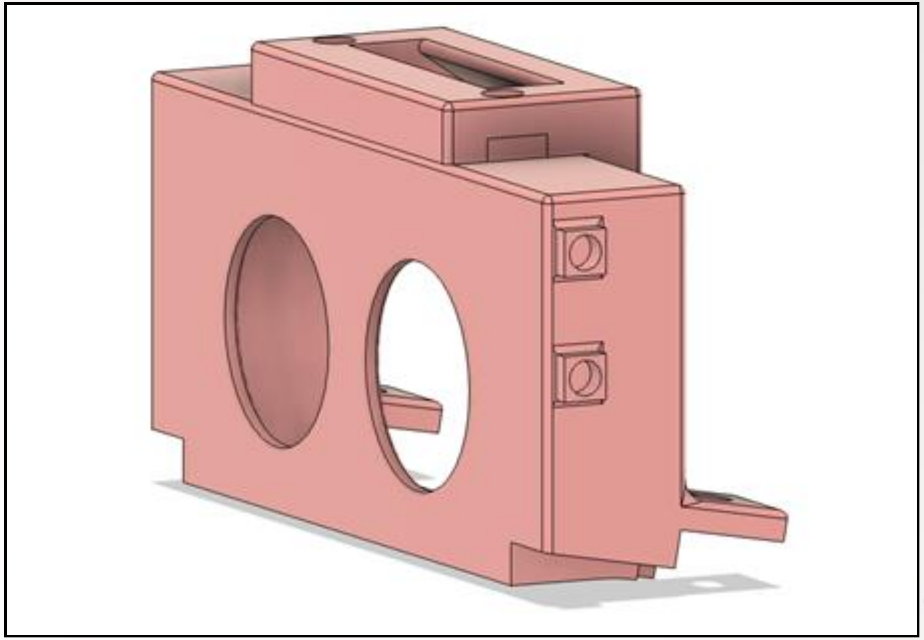
Uses
- 2 M3 heated inserts (on the frame)
- 2 M3x3” screws to hold lip in place



Subcomponent 3: Speaker Cage

USB Speaker is snap-fitted into the Speaker Cage

Speaker Cage is screwed onto the Frame using M3 heated inserts and M3 screws.



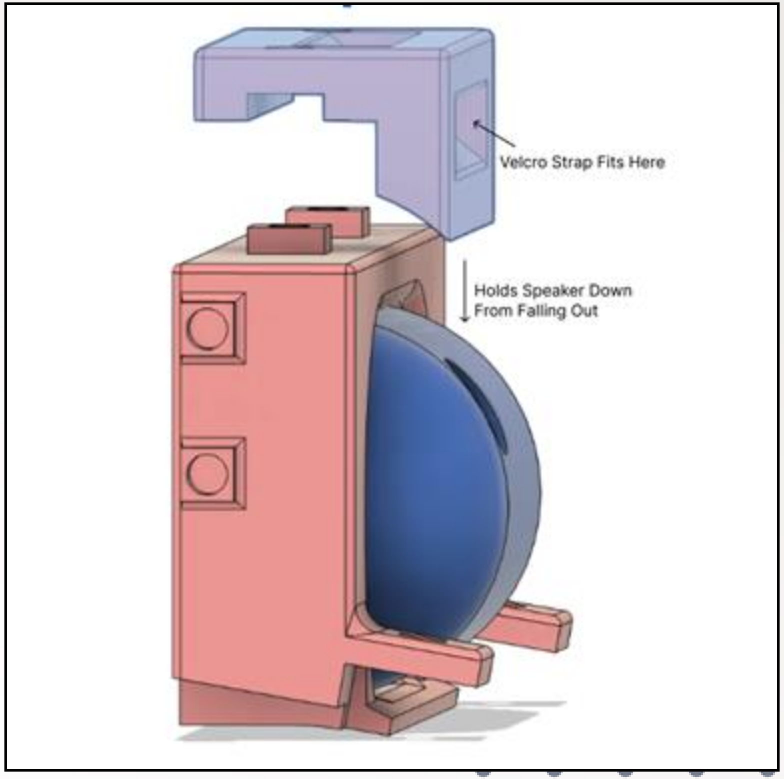
J

Subcomponent 4: Speaker Clamp

Two features

1. Has a rectangular hole for a vertical velcro strap that runs on top of the head
2. Holds the speaker down from falling out of the snap-fitted cage

Speaker clamp is held down by 2 M3 heated inserts and 2 M3 screws

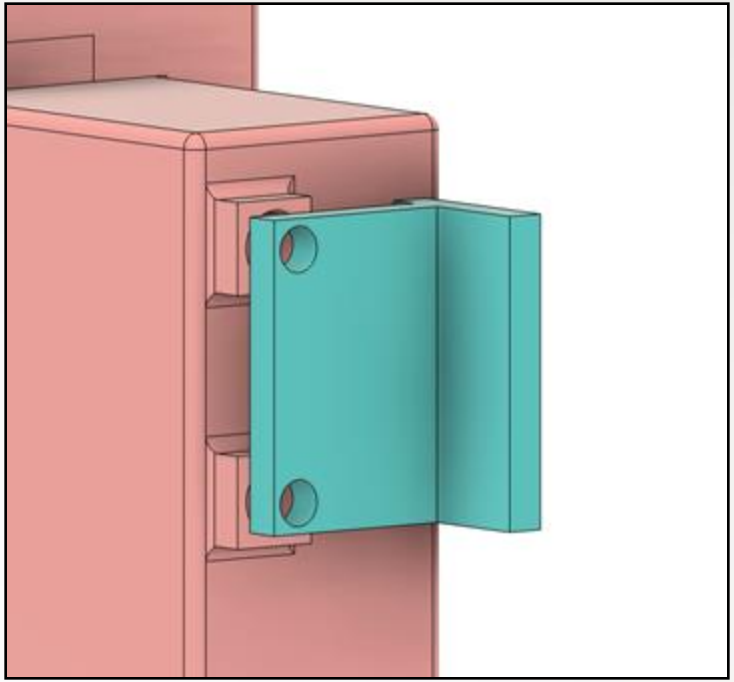


J

Subcomponent 5: Gyro Seat

Additionally, two heated inserts are added to the Speaker Cage

The Gyro Sensor for the headset is screwed down to the Speaker Cage on its side.



J

System Gateway

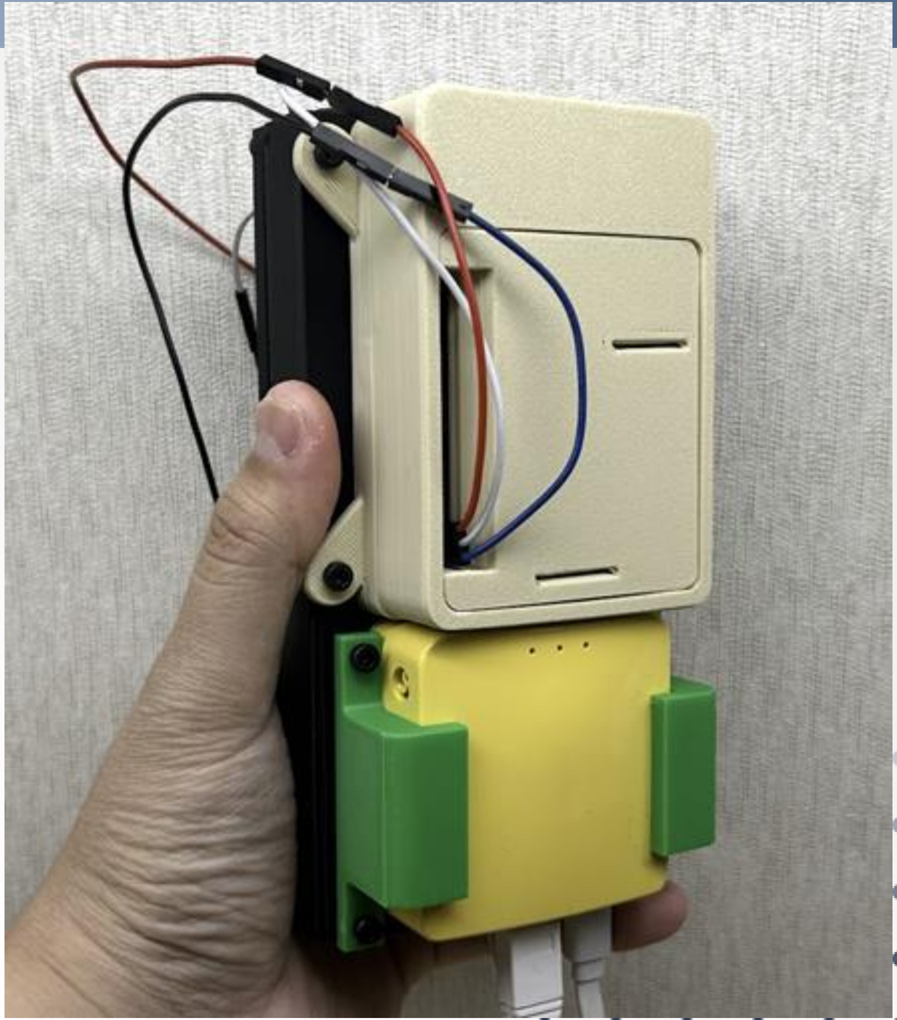
Is physically connected to the Headset's components (Camera, Gyro Sensor, and USB Speaker)

Communicates with the Smart Cane

Runs the core business logic and makes critical decisions

Consists of:

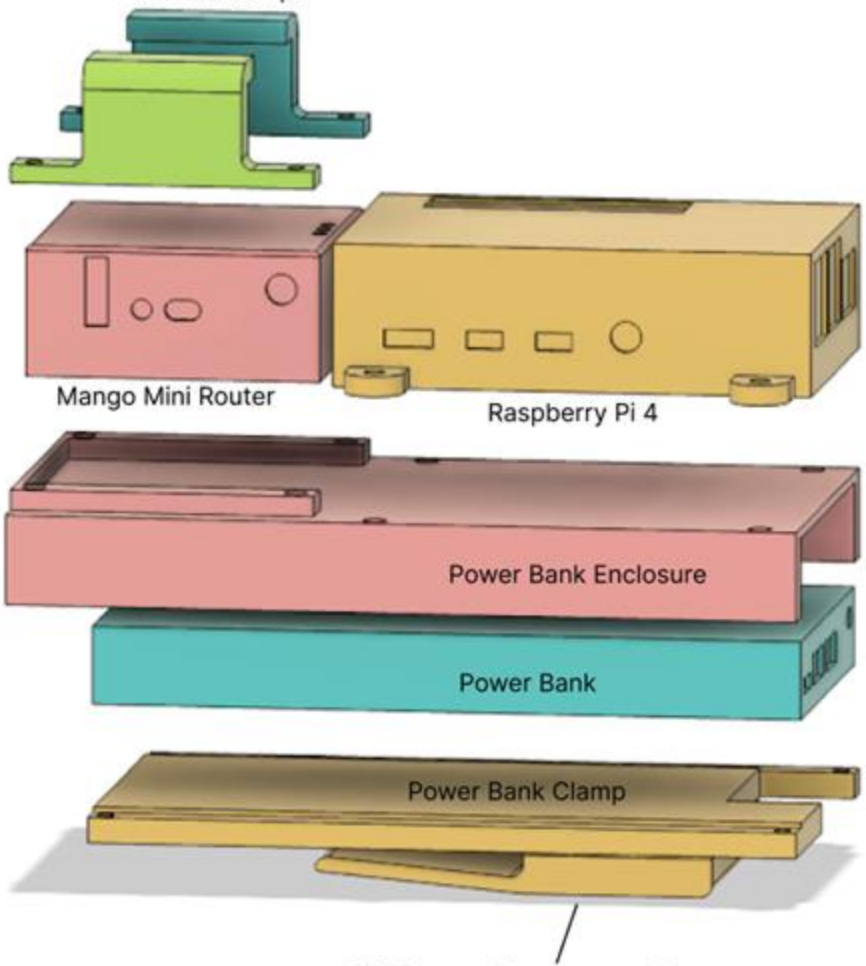
- Raspberry Pi
- Router
- Power Bank



System Gateway: Exploded View

Most items are either clamped or screwed using M3 heated inserts, screws, and nuts.

A belt clip is there in case you want to put it on your belt.

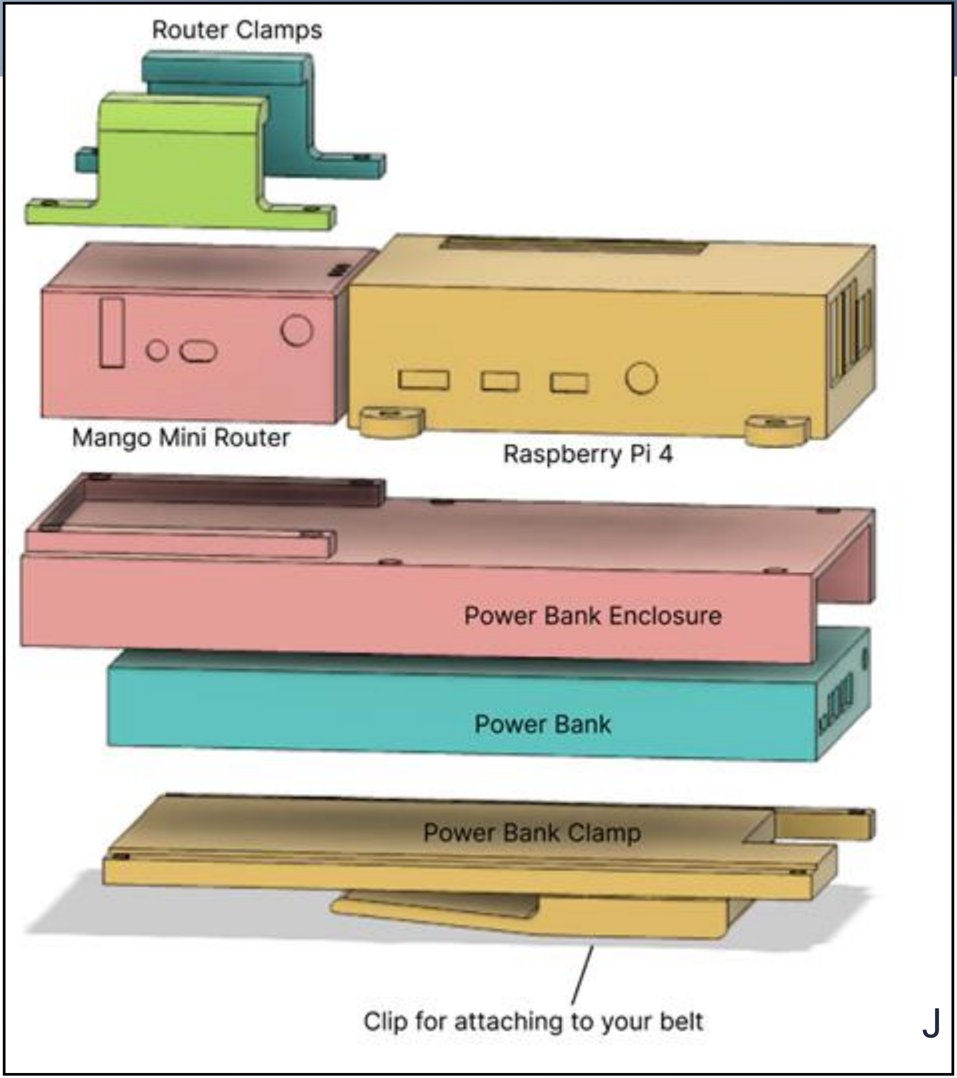


J

System Gateway Subcomponents

Design Subcomponents:

1. Mango Router Clamps
2. Power Bank Enclosure
3. Power Bank Clamp

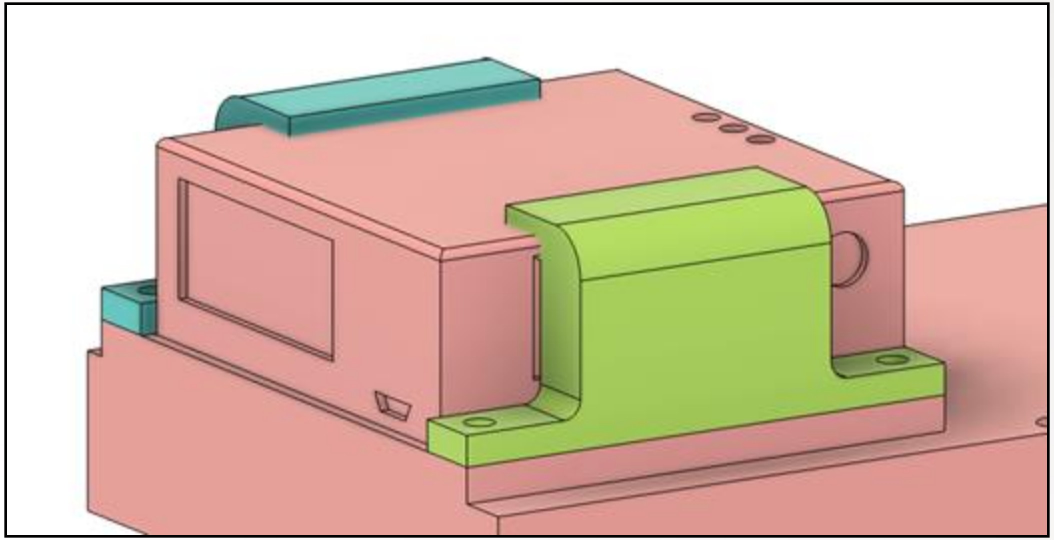


J

System Gateway Design Subcomponent 1

Holds the Router down in place

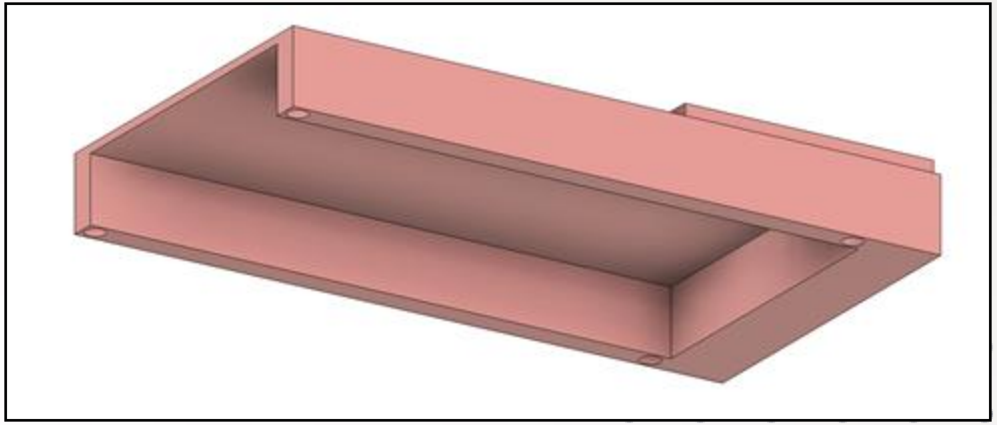
Ensures no important ports are blocked



System Gateway Design Subcomponent 2

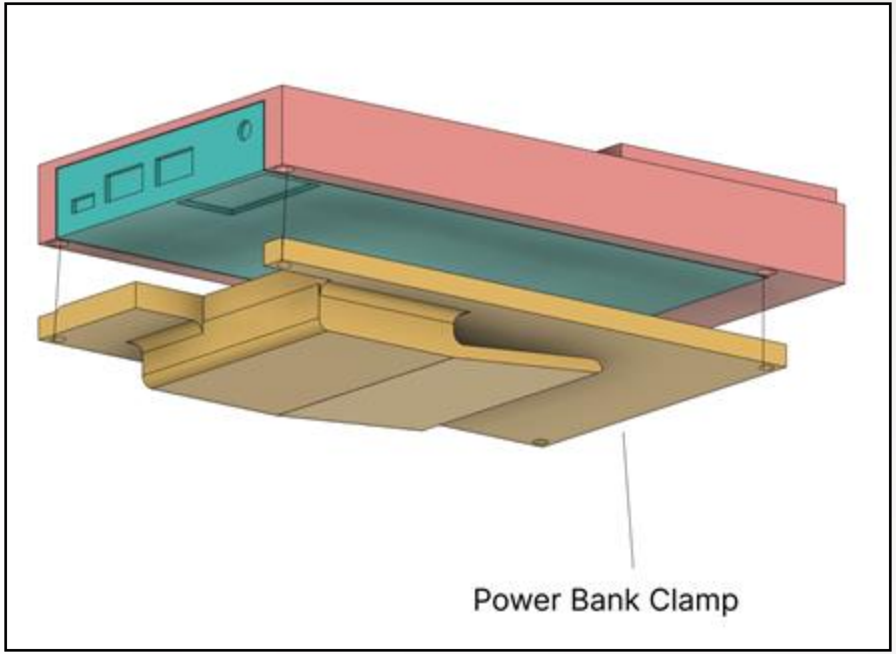
Power Bank Enclosure holds the power bank, with tolerance 0.1mm

Has heated inserts for various other parts to attach themselves to.



System Gateway Design Subcomponent 3

The Power Bank Clamp clamps the power bank to the enclosure, and offers a clip so it can be tucked away nicely on a pants, belt, or bag.



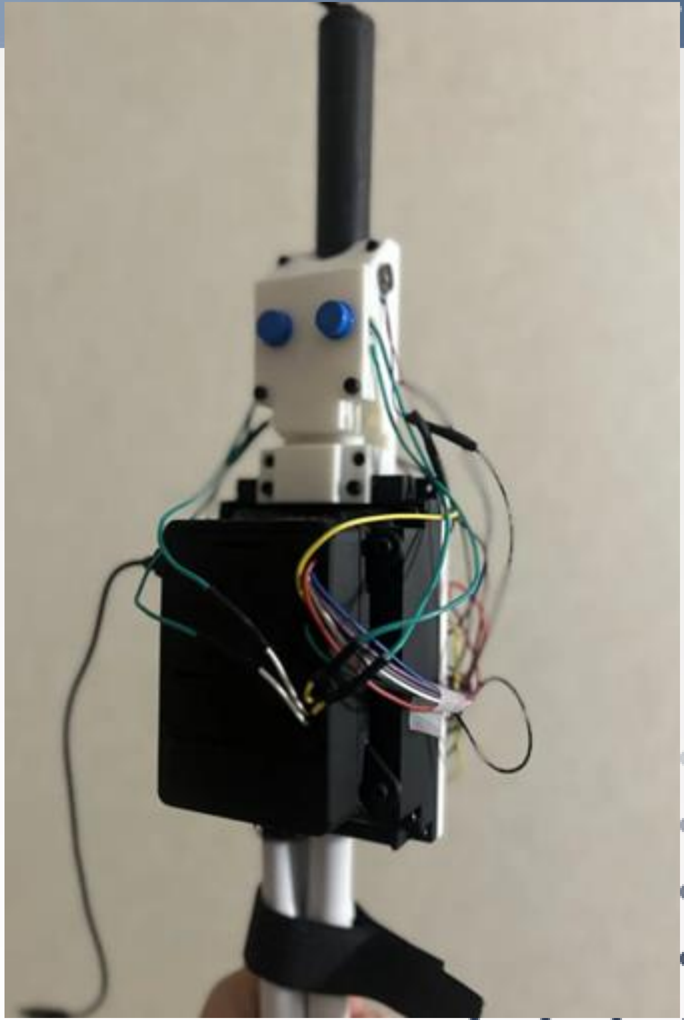
J

Smart Cane

Utilizes a standard White Cane as its base.

Acts as an interface for the user to interact with the system

Contains
- Raspberry Pi
- Push Buttons
- Gyro Sensor
- Relay Modules
- Vibration Motors
- WAGO Terminal Connectors (5V and GND)



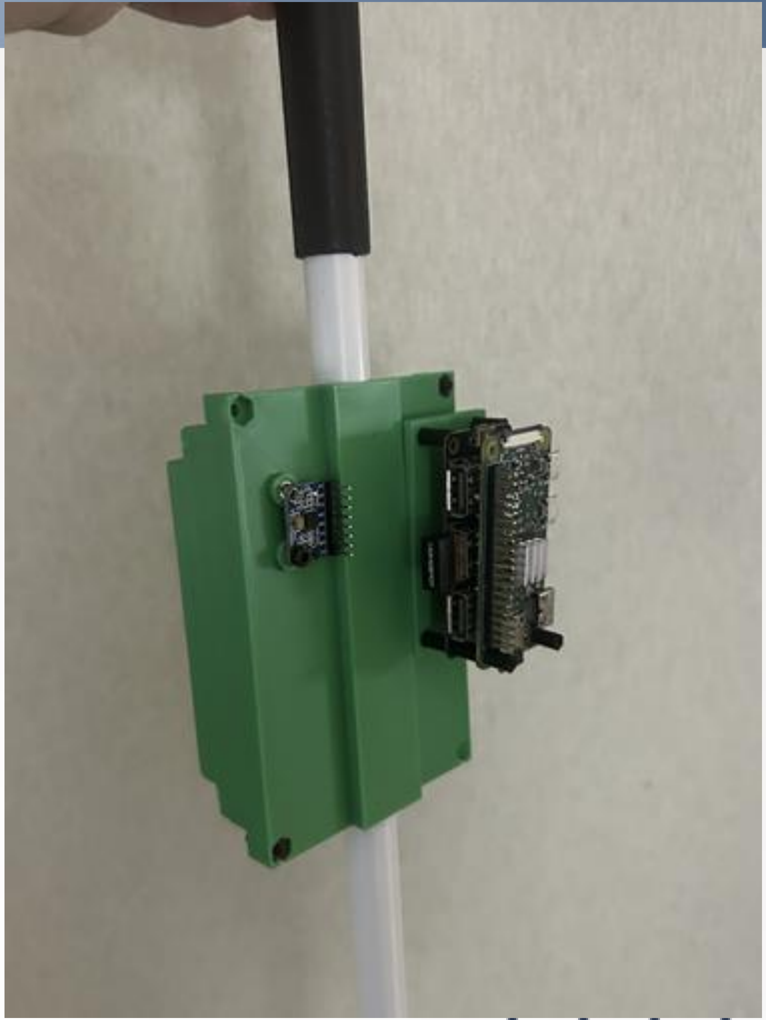
Smart Cane Iteration 1

Goods:

- Had a Gyro Seat
- Had a Raspberry Pi Zero holder
- Had a Power Bank Clamp

Bad

- No Button Interface
- Would Spin with the Cane



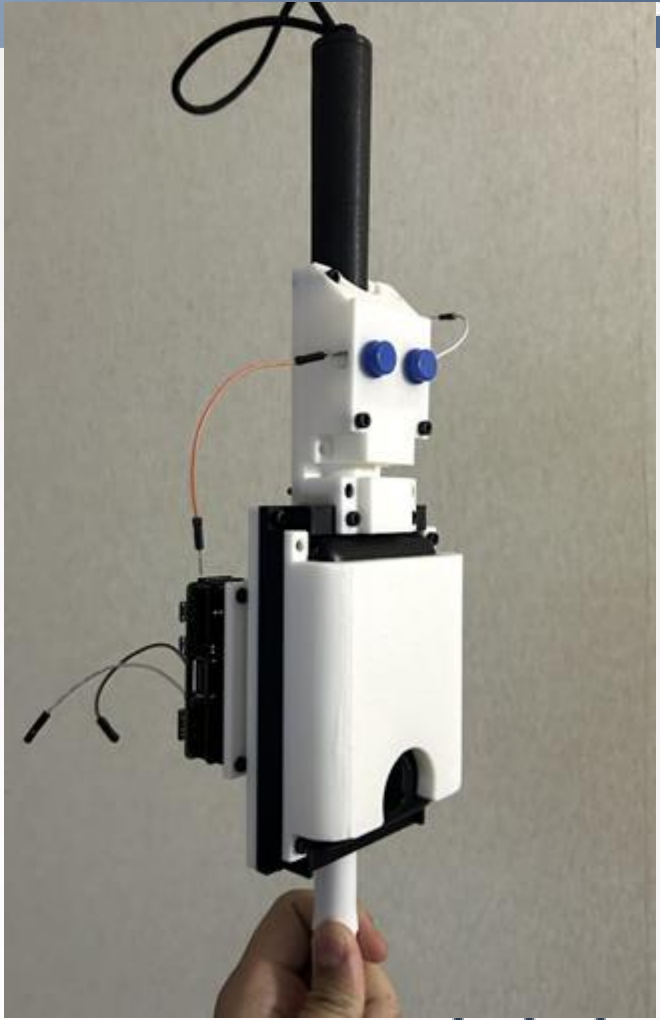
Smart Cane Iteration 2

Improvements Made:

- Added a Button Interface
- Power Bank Clamp Hole was added so that you can view the battery %
- The Electronics Holder component is attached to the button interface, which is fixed, so now they are both fixed and will not spin.

Future Improvements:

- Needs a regular Raspberry Pi to use I2C with Gyro Sensor
- A place for Relay Modules to control the Vibration Motors

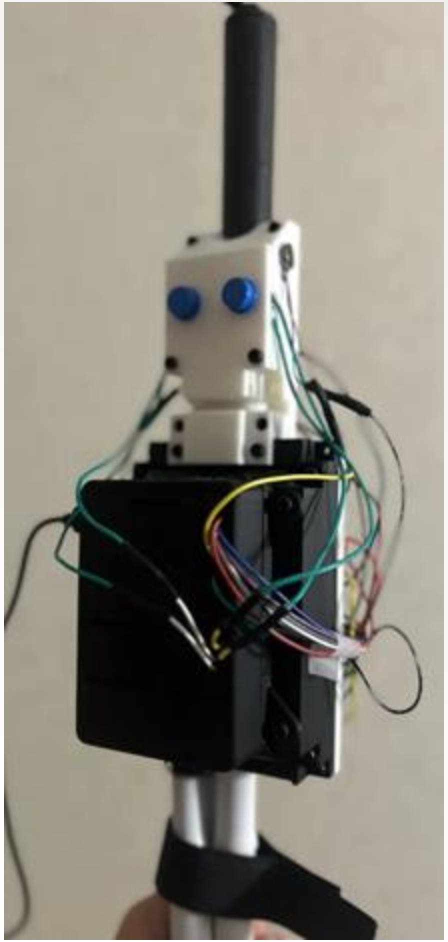


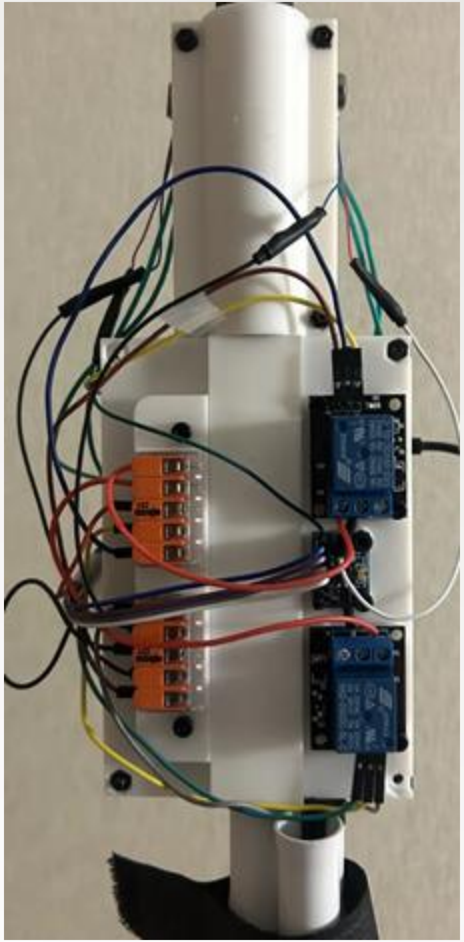
J

Smart Cane Iteration 3 (Final)

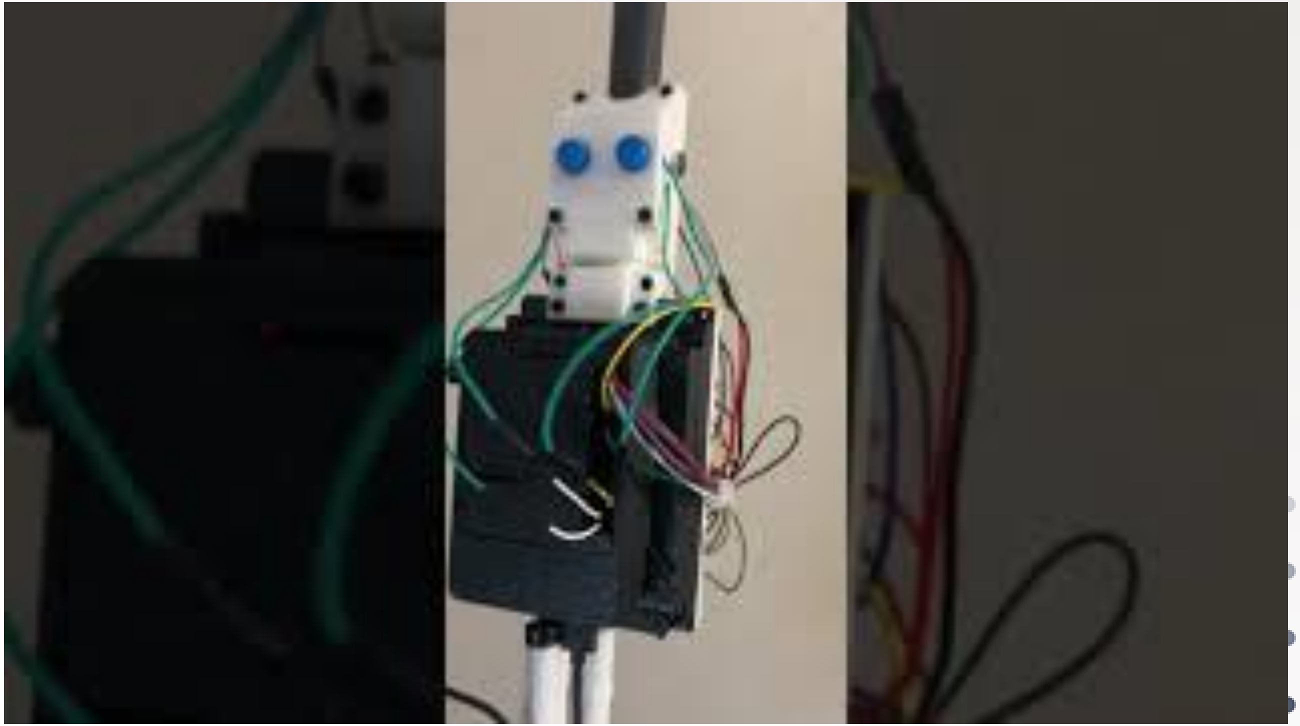
New Features:

- Added Relay Modules
- Added WAGO Connectors to act as terminals for 5V and GND
- Added a Raspberry Pi 4 mount on top of Power Bank Clamp





Smart Cane Iteration 3 (Final)
360 Video

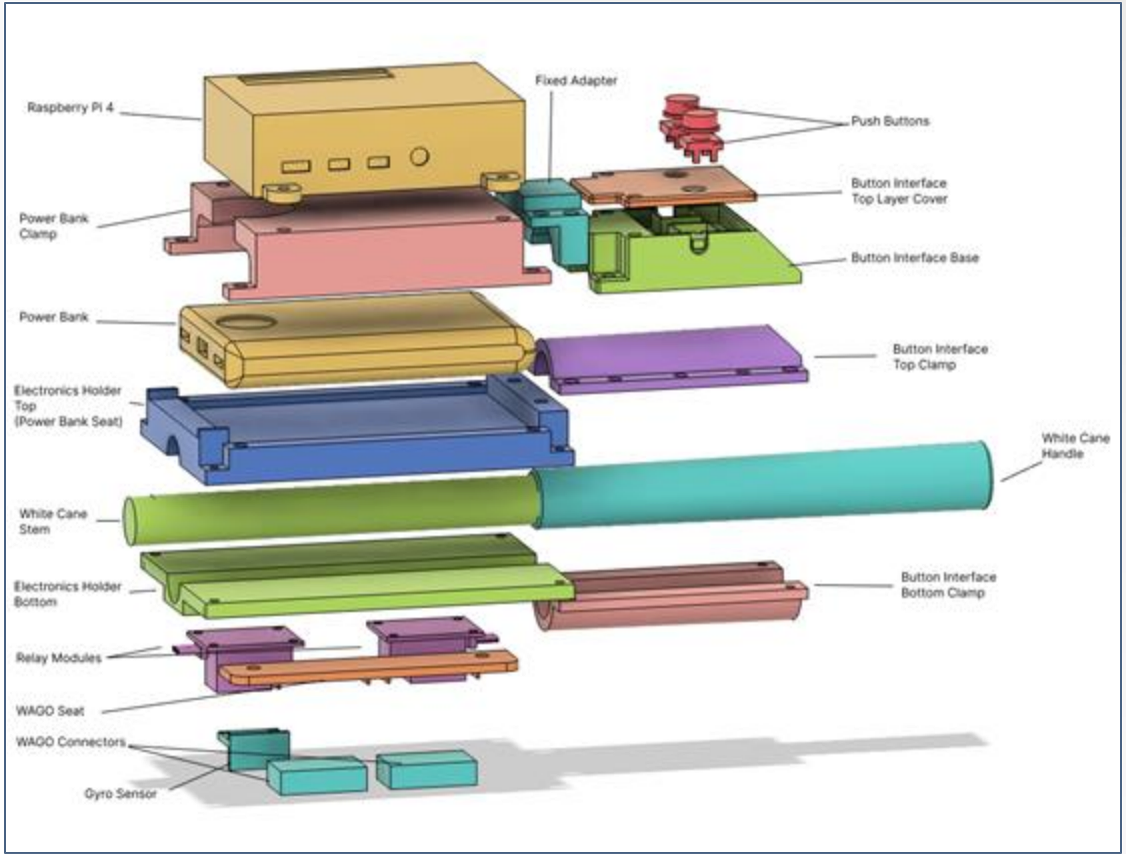


Smart Cane Exploded View

Parts are held together with M3 heated inserts, nuts, and screws.

3 Design Subcomponents:

1. Button Interface
2. Adapter
3. Electronics Holder



J

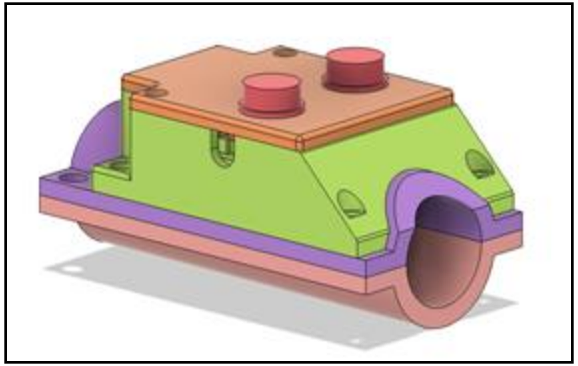
Smart Cane Design Subcomponent 1

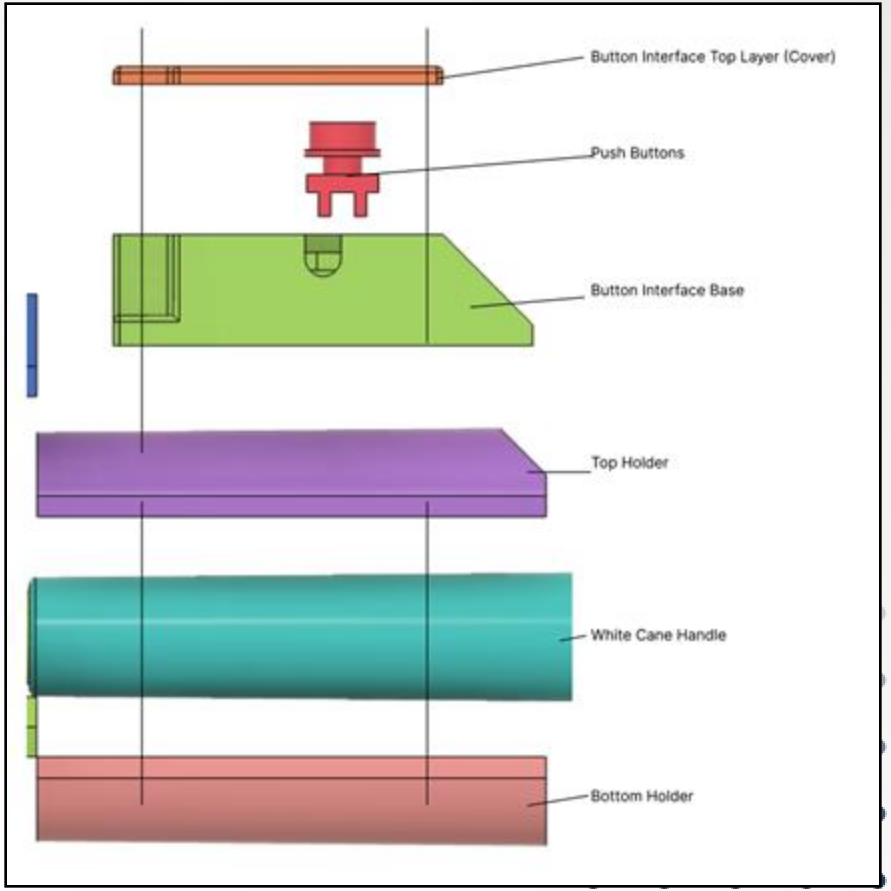
Holds the button in place

Is designed to be ergonomic and easily accessible by the user

Held with m3 heated inserts, screws, and nuts.

The Vibration Motors are also added to the button interface base.





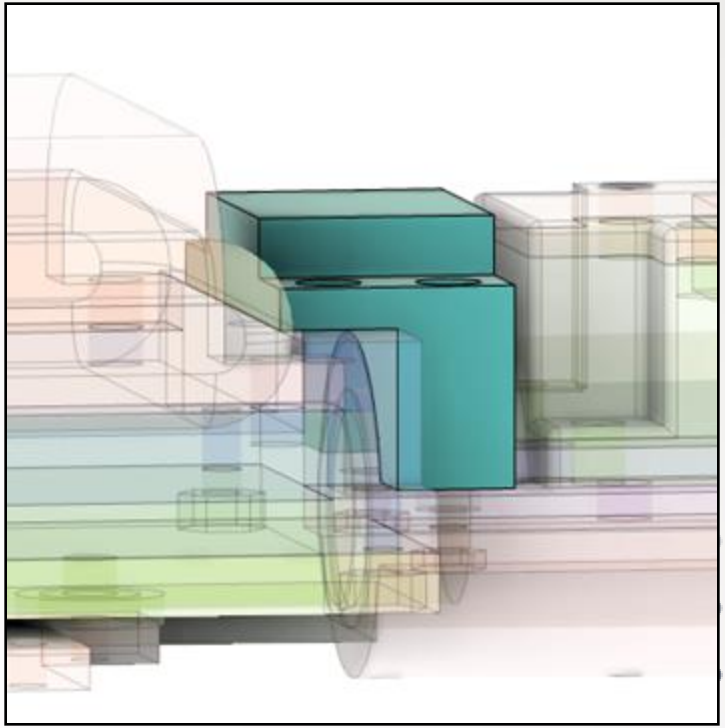
J

Smart Cane Design Subcomponent 2

The adapter connects the two major subcomponents together:

1. The button interface (previous slide)
2. The electronics holder (next slide)

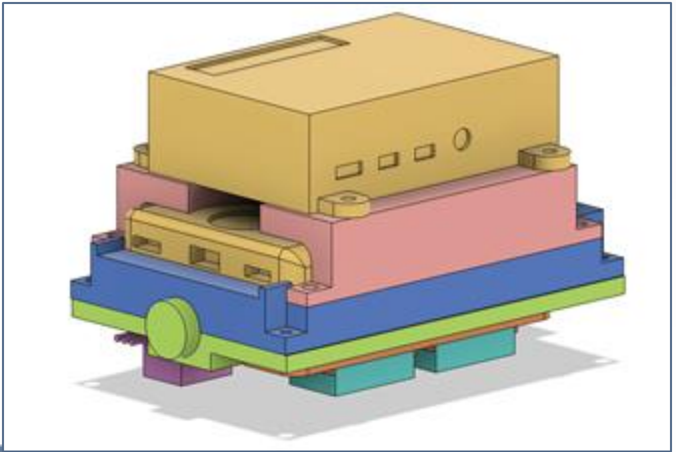
It ensures that the electronics holder is in a fixed position and does not spin along the white cane axis.

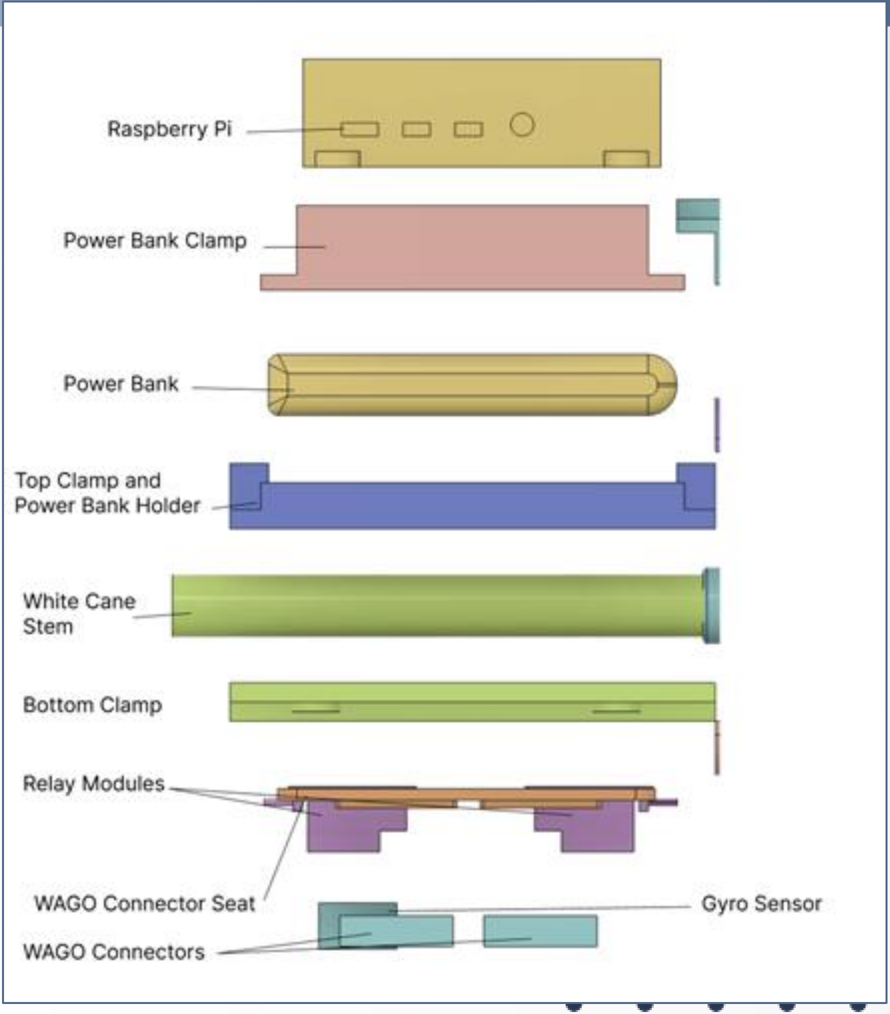


Smart Cane Design Subcomponent 3

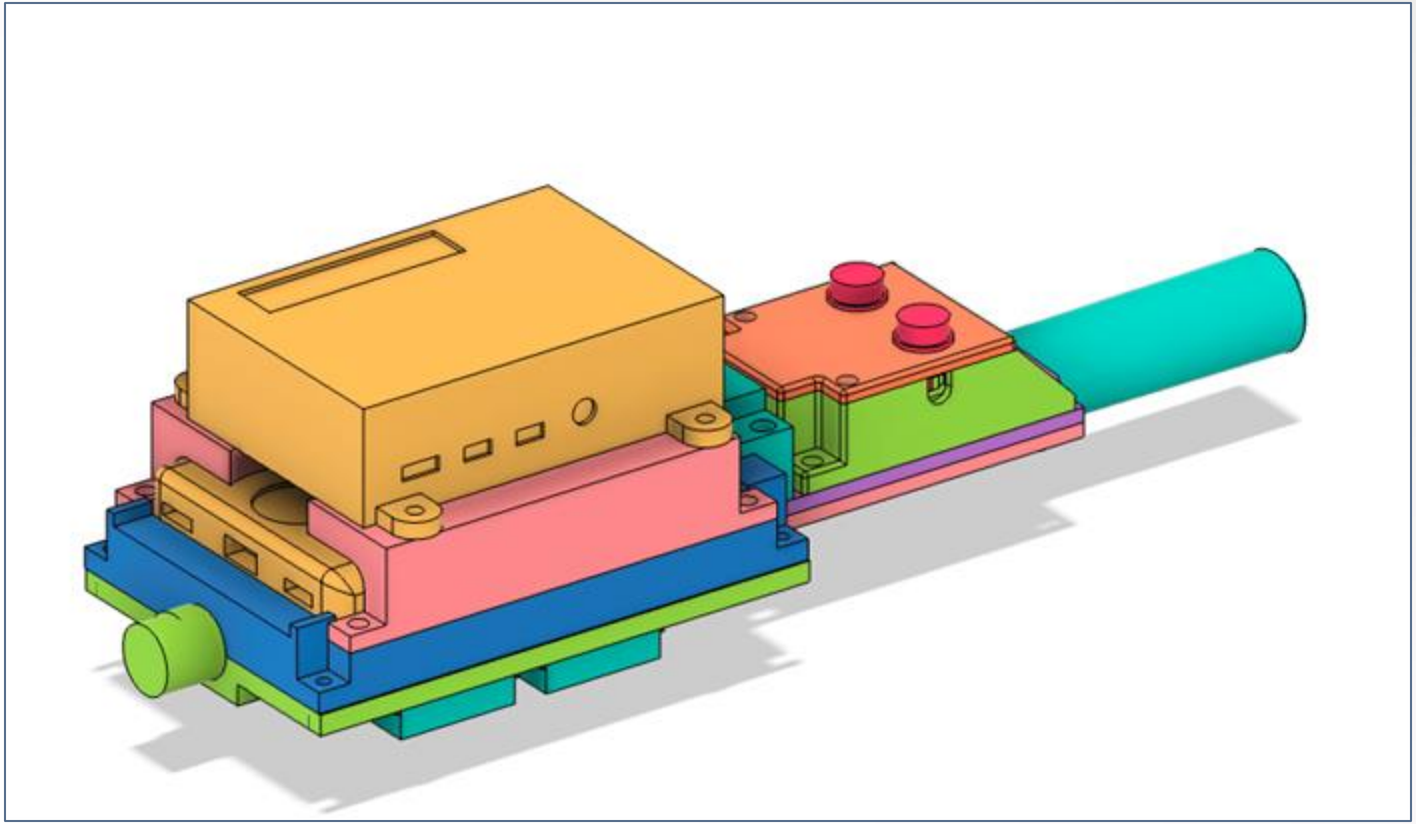
The **Electronics Holder** houses the core electronics for the Smart Cane such as the:

- Raspberry Pi
- Power Bank
- Relay Modules
- WAGO Connector Terminals

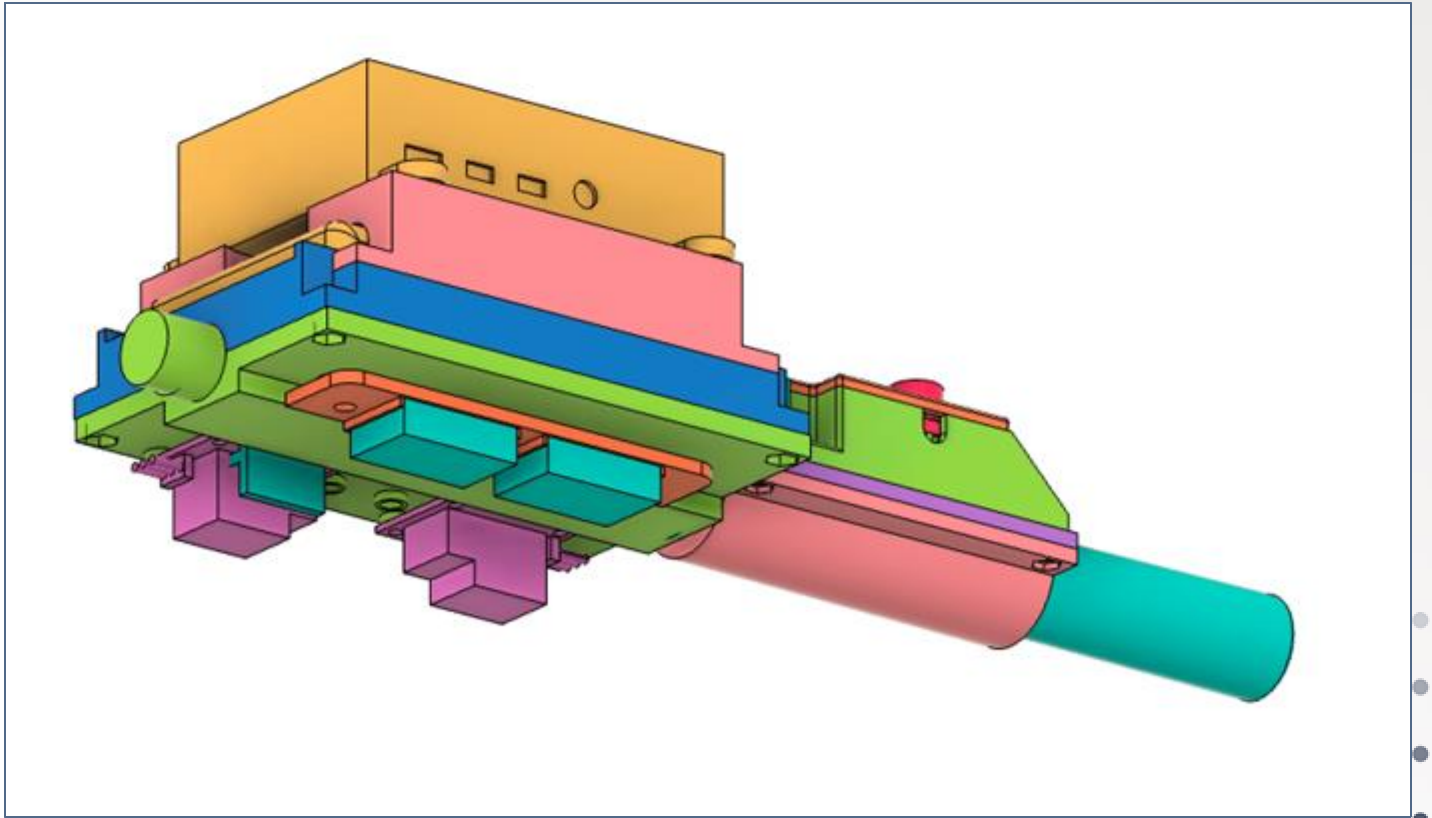




Smart Cane Full Assembly Screenshot 1



Smart Cane Full Assembly Screenshot 2



Software Modules

Model Training

A jupyter notebook for training the model and converting it into a format compatible with the Luxonis Oak-D Lite

System Gateway Raspberry Pi

Contains core business logic with critical decision making

Receives Object Detections and Classifications from Camera

Generates Depth Map

Runs daemon speaker service

Contains lightweight TTS engine

Contains daemon vibration motor controller service

Computes Pitch, Yaw, and Roll of Headset Orientation

Smart Cane Raspberry Pi

A semi-stateless system that does two things:

1. Forwards events to the System Gateway Pi
2. Listens for events sent from the System Gateway Pi

Computes Pitch, Yaw, and Roll of Smart Cane Orientation

Model Training

Collected 214 images of poles, garbage cans, doors, exit signs, general signs, wet floor signs

Manually drew bounding boxes on training images

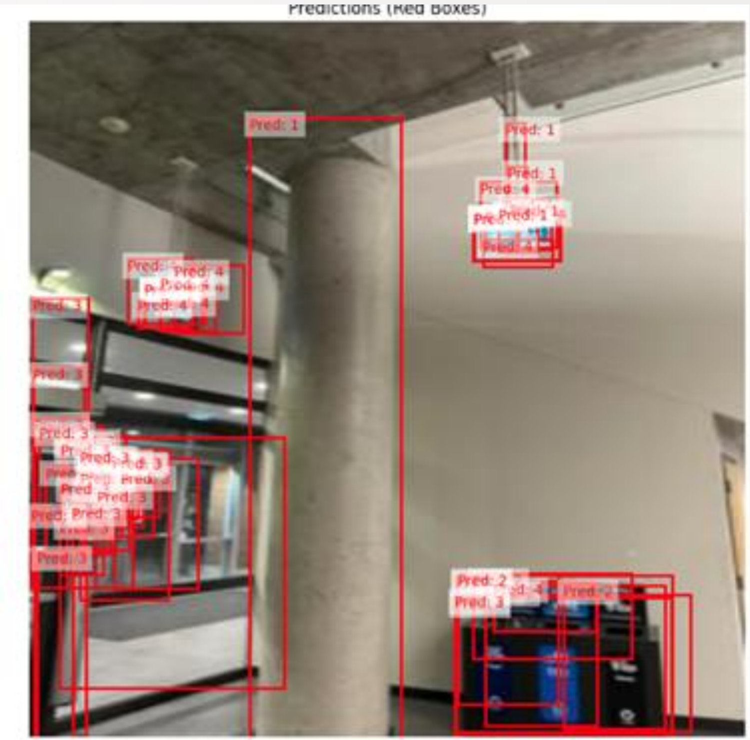
Trained using YOLOv8n as the base model

Converted to `.blob` format which is compatible with Luxonis Oak-D Lite VPU Edge Processing

E

Model Inference Samples

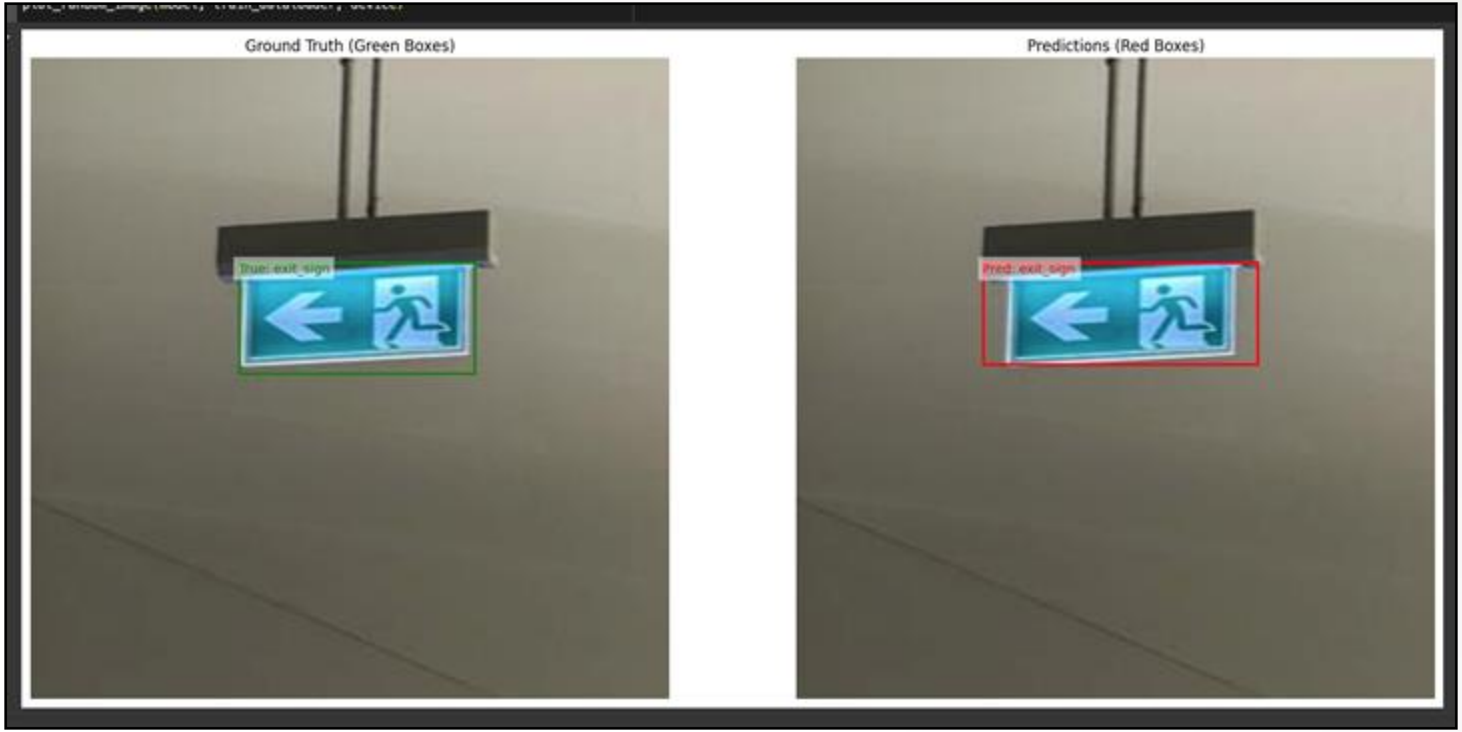
Early iteration of model



E

Model Inference Samples

Final iteration of model



E

UA Walkthrough

E

System Gateway Raspberry Pi

8 Processes running on SystemD services

1. **Camera** → runs pipelines to camera, receives detections and depth map, outputs detections on MQTT
2. **Speaker** → manages volume, plays M4A files, hosts a lightweight TTS engine
3. **Gyro** → reads Gyro values from I2C bus and publishes on MQTT topic
4. **Pitch-Yaw-Roll** → computes Pitch, Yaw, and Roll from Gyro values with Gyro sensor orientation in consideration
5. **Core** → communicates with cane, runs menu options, makes critical decisions
6. **Camera Controller** → exposes HTTP interface to stop, start, restart, and run inference
7. **Vibration Motor Controller** → exposes HTTP interface to run left or right vibration motor on Smart Cane
8. **Dashboard** → exposes a HTML/CSS debug dashboard

J

Smart Cane Raspberry Pi

4 Processes running on SystemD services

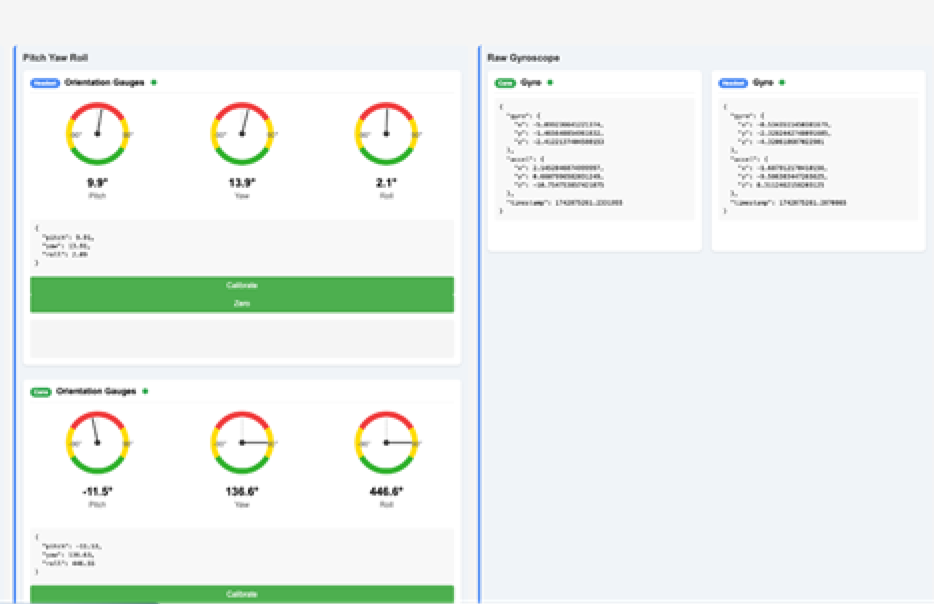
1. **Buttons** → listens for button events and publishes on MQTT topic
2. **Gyro** → reads Gyro values from I2C bus and publishes on MQTT topic
3. **Pitch-Yaw-Roll** → computes Pitch, Yaw, and Roll from gyro values and publishes to MQTT
4. **Vibration Motors** → listens on System Gateway MQTT and actuators vibration motors if instructed to do so.

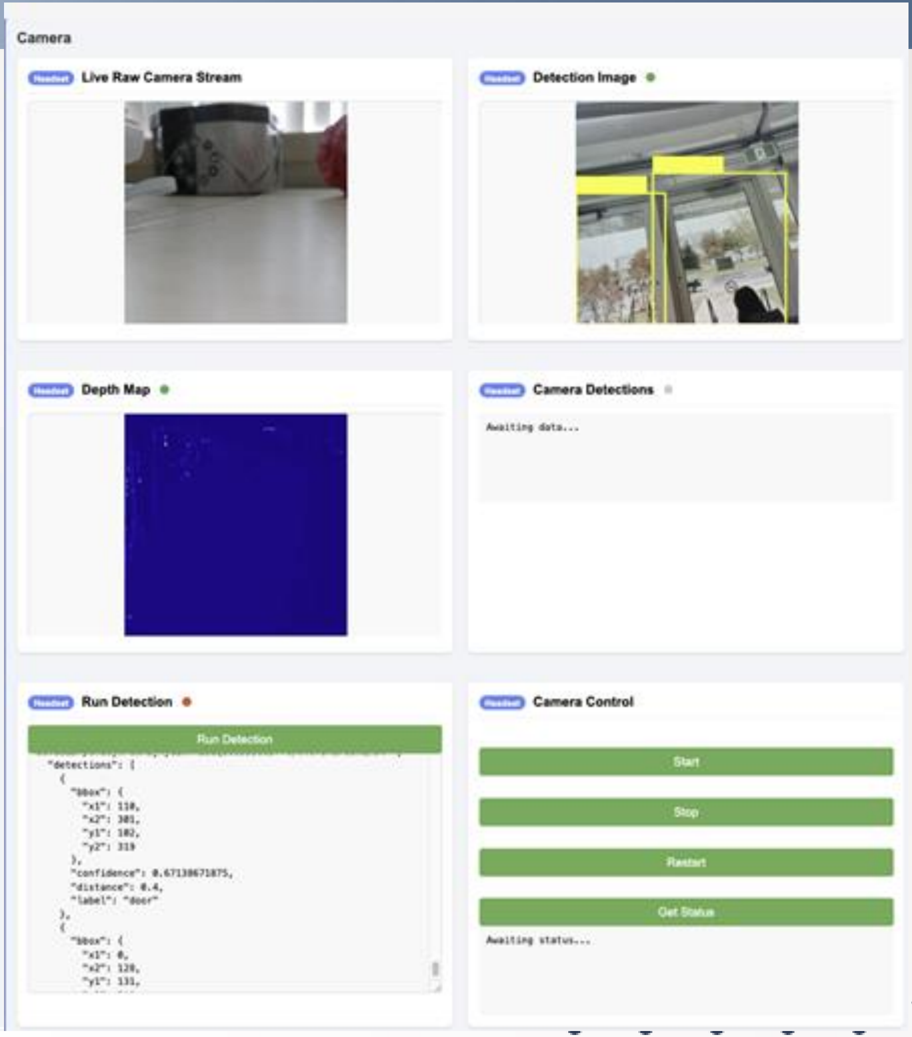
Debug Dashboard

15 different controls
19 different panels to view system status from

Mainly used as a debug dashboard.

Contains: Gyro gauges, depth map, live video feed, speaker controls, camera controls, and manual TTS.

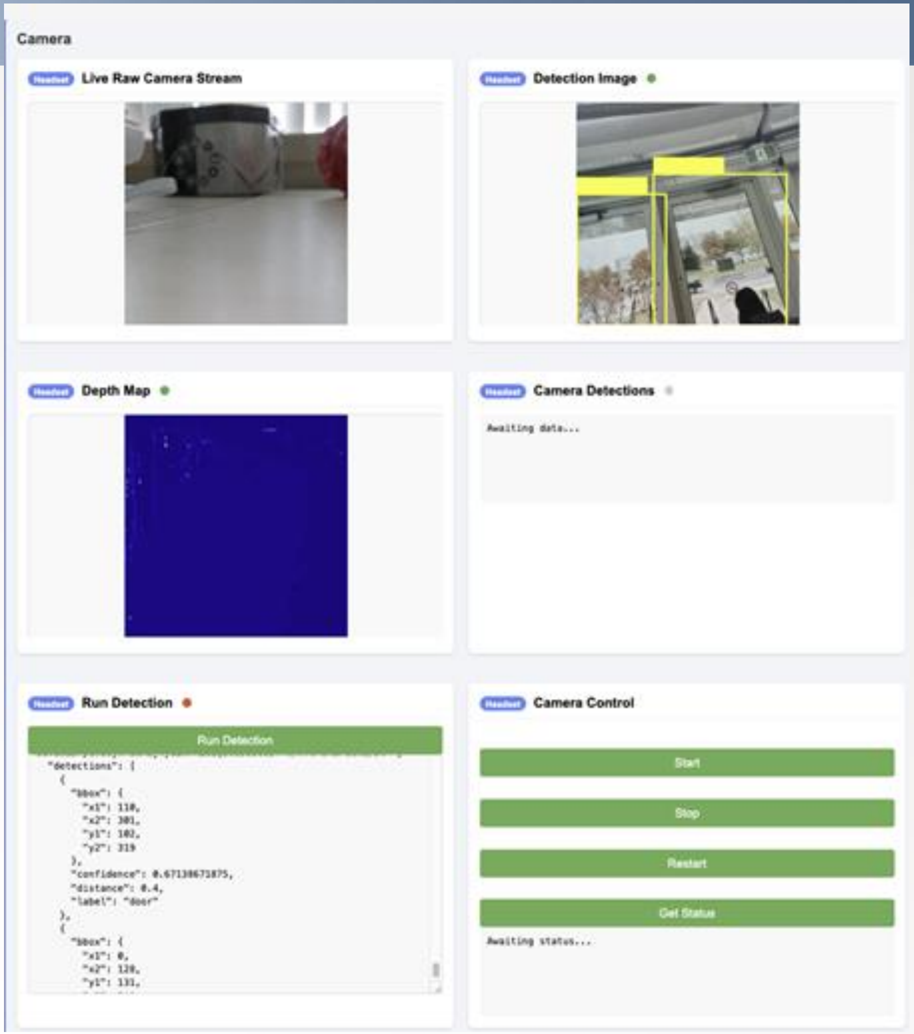




Debug Dashboard

Camera Controls

- View Live Raw Camera Stream
- View the Detection Image (when detection is ran)
- View Depth Map (when detection is ran)
- View list of detections in JSON format
- Run the Detection with a button through HTTP interface
- Start, Stop, Restart the Camera or Get Camera Status



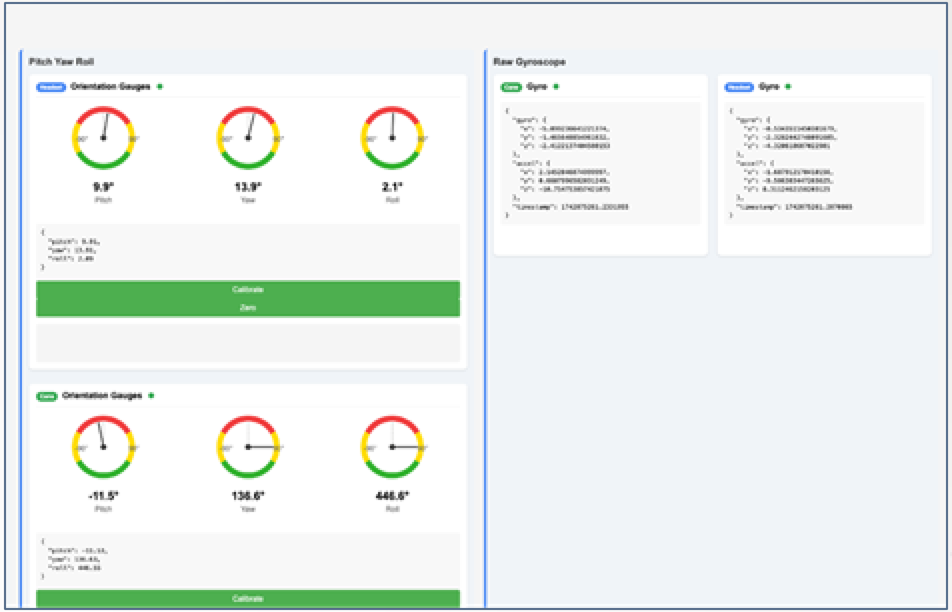
Debug Dashboard


```
  "detections": [
    {
      "bbox": {
        "x1": 110,
        "x2": 301,
        "y1": 102,
        "y2": 319
      },
      "confidence": 0.67138671875,
      "distance": 0.4,
      "label": "door"
    },
    {
      "bbox": {
        "x1": 0,
        "x2": 128,
        "y1": 131,
```

Debug Dashboard

Gyro Controls

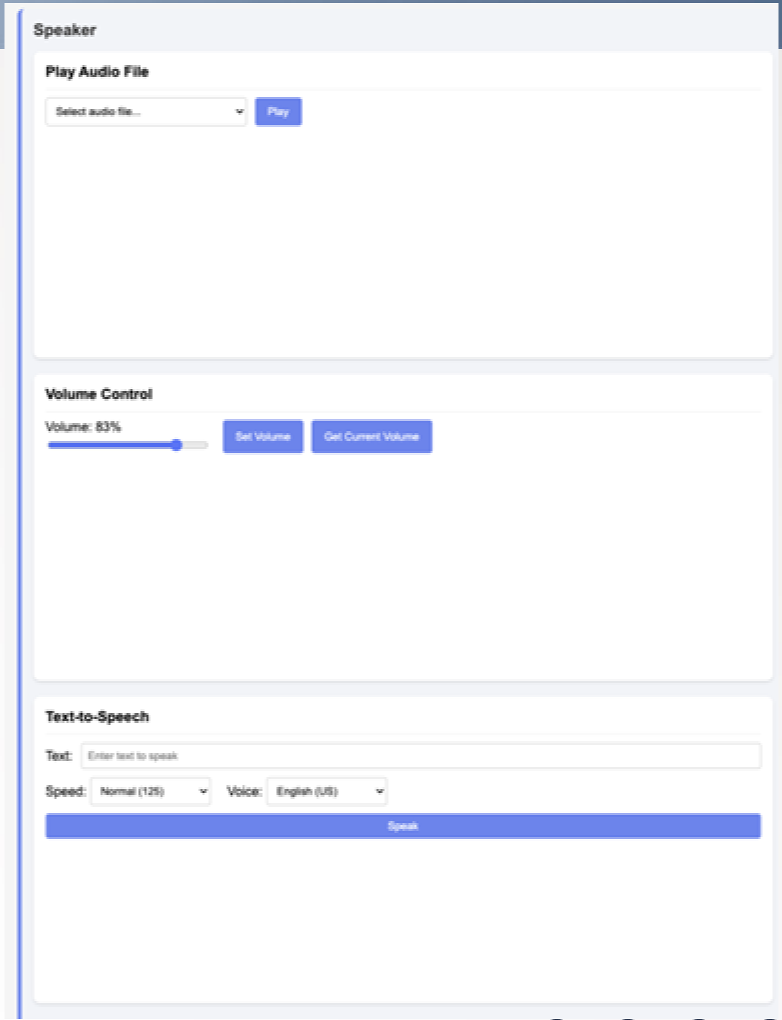
- Headset Raw Gyro Values (Gyro X, Y, Z, Acceleration X, Y, Z)
- Smart Cane Raw Gyro Values (Gyro X, Y, Z, Acceleration X, Y, Z)
- Headset Pitch, Yaw, Roll Gauges
- Smart Cane Pitch, Yaw, Roll, Gauges
- Headset Calibrate Pitch, Yaw, Roll
- Smart Cane Calibrate Pitch, Yaw, Rol,
- Headset Zero Yaw
- Smart Cane Zero Yaw



Debug Dashboard

Speaker Controls

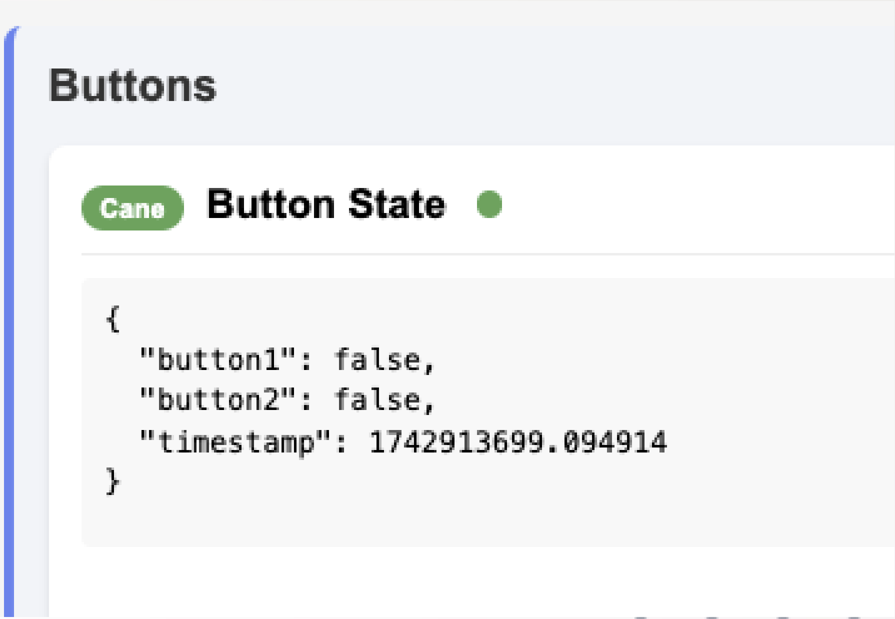
- Play a pre-downloaded audio file (like a Notification Sound)
- Set Volume (Slider)
- Get Volume
- Enter Text for the TTS Engine to Narrate
    - Select Speed
    - Select Voice Language/Accent



Debug Dashboard

Button State

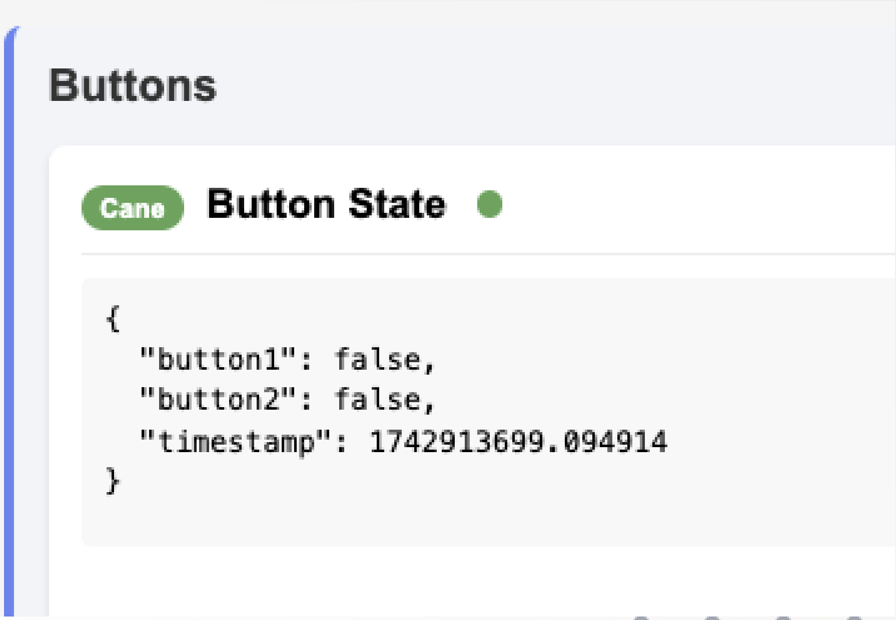
- View the button state MQTT topic live from the Smart Cane
- Used primarily by `core` to make effective decisions



Debug Dashboard

Button State

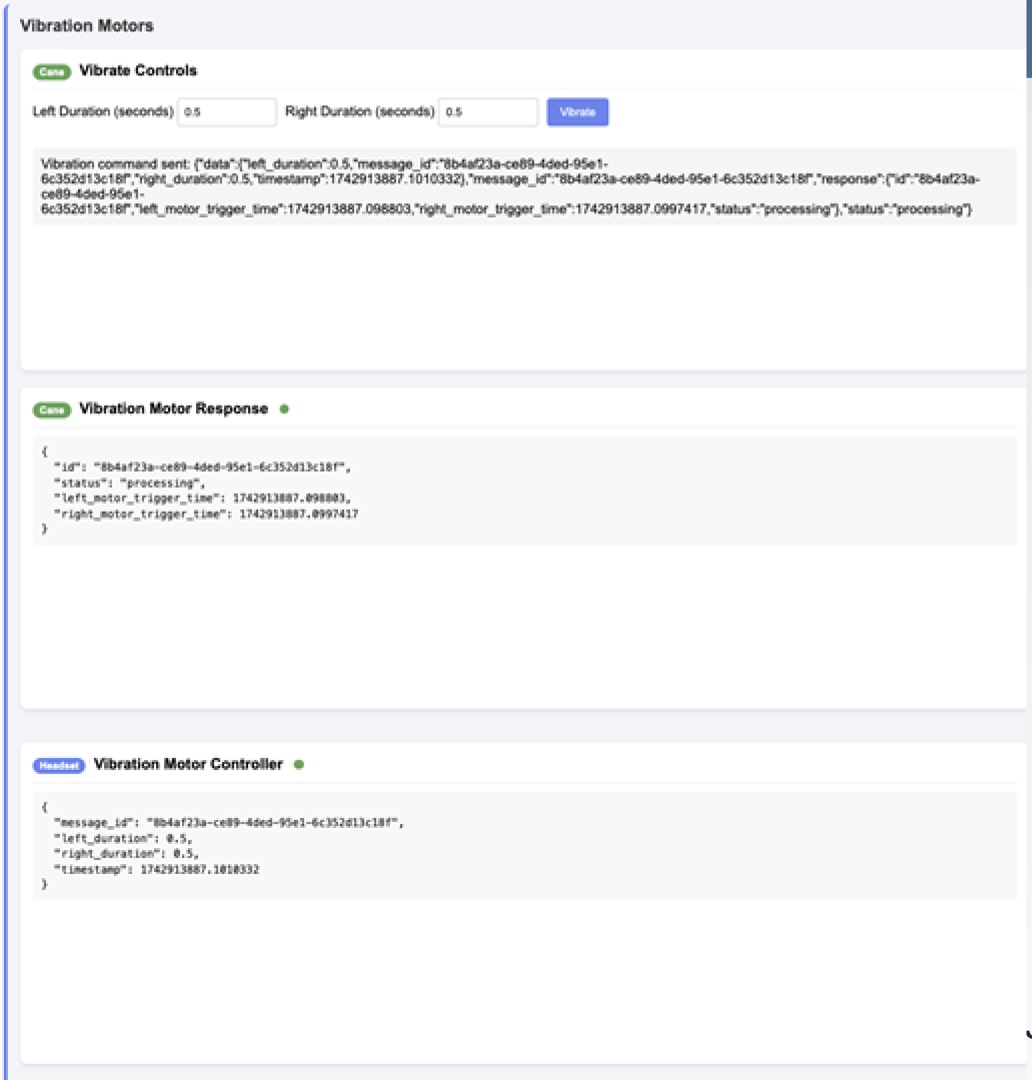
- View the button state MQTT topic live from the Smart Cane
- Used primarily by `core` to make effective decisions



Debug Dashboard

Vibration Motor Controller

- Vibrate Motors
  - Left for X Seconds
  - Right for X Seconds
  - Sends HTTP Request to Controller
- Controller publishes a message on their MQTT Topic on Headset Pi
- Smart Cane is listening to MQTT Topic and sends an ACK response
- Smart Cane actuates Vibration Motors through Relay Modules



J

Use Case 1: Obstacle Detection & Classification

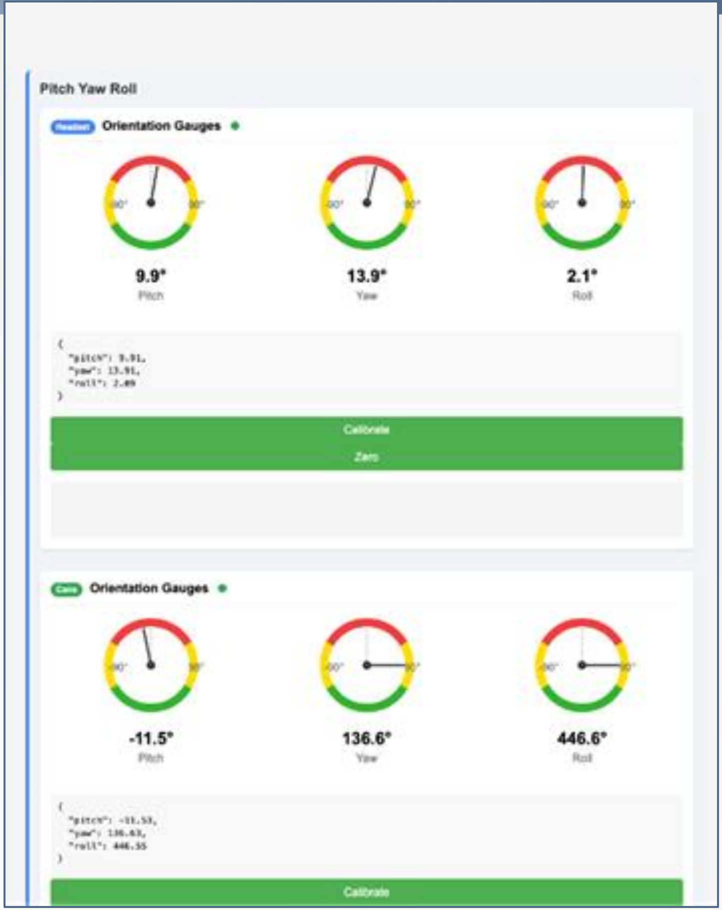
1. Camera Detects Objects
2. Speaker Service Narrates The Detections and their Distances


```
"detections": [
  {
    "bbox": {
      "x1": 110,
      "x2": 301,
      "y1": 102,
      "y2": 319
    },
    "confidence": 0.67138671875,
    "distance": 0.4,
    "label": "door"
  },
  {
    "bbox": {
      "x1": 0,
      "x2": 128,
      "y1": 131,
```


J

Use Case 2: Object Tracking

1. Camera Detects Objects
2. Speaker Service Narrates The Detections and their Distances
3. User Chooses Which To Track
4. Gyro Sensors Calculate Angle Offset
5. Left/Right Vibration Motors Guide The User To Object



Use Case 3: Text Scanning

1. Camera Detects a Sign
2. User Walks Up To a Sign
3. User Activates Text Scanning
4. AWS Textract is used to scan sign
5. Speaker TTS Narrates Text

J

3

Testing Approach and Results

A

Testing Approach

OUR AIM

Our aim was to test components individually and when they were integrated together.

OUR CRITERIA

Expectations of our tests were based on research for stakeholders and on professor's suggestions

Hardware

Hardware Components: Luxonis Oak-D Lite, Mango Router, Power Bank, Raspberry Pi 5 8 GB, Raspberry Pi 4 2 GB, USB Speaker, Push Buttons, Relay Module, Vibration Motors, Gyro Sensor

A

Acceptance Testing Process

What We tested for:
We tested for usability, operation duration, and properly working features

Procedure
A different methodology is used for each kind of test

Manual tests are conducted when doing functional tests and results are observed by the tester

Usability tests are completed in a controlled environment, where the settings are in an indoor environment

Performance testing is tested using a testing script

Acceptance Testing Criteria

Stakeholder Testing: The product is tested against stakeholder requirements to confirm that it meets user expectations and fulfills the intended purpose.

Functional Testing: Each system component is verified against functional specifications, ensuring that all features, workflows, and business processes operate correctly. This includes validating data processing, calculations, and system integrations.

Usability Testing: Real users interact with the system to assess its intuitiveness, accessibility, and ease of use. Feedback is gathered on learnability and user satisfaction.

Compatibility Testing: The system is tested across different browsers, operating systems, devices, and network configurations to ensure consistent performance in all intended environments.

Performance Testing: The system's responsiveness, stability, and resource utilization are evaluated under various conditions, including normal usage, peak loads, and stress scenarios. This ensures it meets performance benchmarks for speed, scalability, and reliability.

User Scenario

User Scenarios	Related Tests
A visually impaired user is navigating through the interior of Durham College, where a support pillar is located in the center of one of the pathways. As they approach a pole, the system detects it and provides a visual bounding box with a distance estimate.	S-1
A visually impaired user is navigating through the interior of Durham College, where a support pillar is located in the center of one of the pathways. As they approach a pole, the system detects it and announces the object detected, in this case, a pole, and the distance through the headset speaker. They are also able to gauge the direction through the haptic feedback on the cane.	S-2, S-3
A system administrator opens the GUI to monitor the Smart Assistant's performance, viewing live object recognition and adjusting settings as needed.	S-4

A

Stakeholder Testing

These tests validate whether the system’s features function correctly in practical scenarios, and meet real word expectations of users/stakeholders

Test Case #	Test Name
S-1	Object Recognition and Object Distance Estimation
S-2	Text to speech functionality and speaker functionality
S-3	Gyro Sensor Connections
S-4	GUI for the system

Functional Testing

15 Test Cases

Functional testing ensures that each component of the Smart Assistant system operates according to the defined functional specifications. Verifies that all features, workflows, and integrations behave as expected. Most are to test software execution

Test Case #	Test Name	Description	Preconditions	Postconditions
A-1	Announce Detections	Read detections that are inferred from the camera and use the speaker service's Text To Speech capabilities to narrate the detections to the user	Camera is on. Objects detected. Speaker service fully functional	Detections are narrated to the user

Usability Testing

Usability testing evaluates how intuitive and user-friendly the Smart Assistant system is for its intended audience. This process assesses ease of navigation, accessibility, and overall user experience by observing real users interacting with the interface.

Test Case #	Test Name	Description	Preconditions	Postconditions
B-1	White Cane Button Controls Said on Speaker	Whenever a button is pressed on the white cane, the speaker will speak out what option that button does	All components of the white cane assistant are turned on	Text-to-speech command is issued to speaker
B-2	Cane is still usable	The system is usable and intractable without the need for vision	Have the cane components physically on the cane	Able to use the cane as normal
B-3	System is usable without vision	The system should be usable by the blind and visually impaired	Be able to interface with the system with your eyes closed	Be able to use the system

Compatibility Testing

Make sure program is compatible with pre-existing hardware or software

Test Case #	Test Name	Description	Preconditions	Postconditions
C-1	Dashboard on different web browsers.	Test that the debug dashboard works on different web browsers	Dashboard is operational	Dashboard is responsive and functional

Performance Testing

Tests on if the blind person assistant system has acceptable reliability, speed, and responsiveness.

8 Tests

Most Tests are tested using the Test.sh script

By testing, we found the limitations of the of running the AI assistant program for too long.

Test Case #	Test Name	Description	Preconditions	Postconditions
D-1	Headset Battery Life (Idle)	Tests the battery life capabilities under idle load	All systems and components are powered and functional in the headset, but no continuous usage. Battery is charged at 100% before test	Battery operates as long as possible

A

Test.sh

```
Execution time: 0.5 seconds
jeremy@raspberrypi4:~/headset $ cat test.sh
#!/bin/bash

if [ $# -eq 0 ]; then
    echo "Usage: $0 command [arguments]"
    echo "Example: $0 ls -la"
    exit 1
fi

cmd="$@"

echo "Executing: $cmd"
echo "----------------------------------------"

start_time=$(date +%s%N)

eval "$cmd"
cmd_exit_code=$?

end_time=$(date +%s%N)

elapsed_ns=$((end_time - start_time))

seconds=$((elapsed_ns / 1000000000))
milliseconds=$(((elapsed_ns % 1000000000) / 1000000))

echo "----------------------------------------"
echo "Command completed with exit code: $cmd_exit_code"
echo "Execution time: ${seconds}.${milliseconds} seconds"
jeremy@raspberrypi4:~/headset $
```

- Testing script that checks the execution time of commands
- Computes the execution time in seconds
- Ran by using connected debug laptop
- All responses are recorded in a second

```
jeremy@raspberrypi4:~/headset $ ./test.sh curl localhost:8010/status
Executing: curl localhost:8010/status
----------------------------------------
{"detail":"Unable to connect to video stream","port_accessible":false,"service_status":"inactive","status":"offline"}
----------------------------------------
Command completed with exit code: 0
Execution time: 0.39 seconds
```

Test Results

All of all test are found to be successful

Example successful cases:

Stakeholder Test Case #	Date Test Performed	Test Owner	Result: Pass/Fail	Comments
S-1	Object Recognition and Object Distance Estimation	Dr. Khalid Abdel Hafeez	Pass	Maybe try to make it recognize corners

Functional Test Case #	Date Test Performed	Test Owner	Result: Pass/Fail	Comments
A-1	March 5, 2025	Jeremy Mark Tubongbanua	Pass	Detections that are outputted by the camera stream service can be successfully narrated by the speaker using the Text To Speech engine in real-time.

Test Results

Usability Test Case #	Date Test Performed	Test Owner	Result: Pass/Fail	Comments
B-1	March 13, 2025	Andy Lau	Pass	No comments
B-2	March 13, 2025	Jeremy Mark Tubongbanua	Pass	Typically, the visually impaired are already comfortable with a cane, so it is already integrated with a tool they are familiar with.
B-3	March 13, 2025	Jeremy Mark Tubongbanua	Pass	Menu navigation is functional fully with the buttons, and do not have to use your eyes in order to interact with the system

Test Results

Compatibility Test Case #	Date Test Performed	Test Owner	Result: Pass/Fail	Comments
C-1	March 13, 2025	Jeremy Mark Tubongbanua and Emily Lai	Pass	Dashboard is responsive and functional on Google Chrome, Safari, Firefox, and also on mobile phones.

A

Test Results

Performance Test Case #	Date Test Performed	Test Owner	Result: Pass/Fail	Comments
D-1	March 13, 2025	Jeremy Mark Tubongbanua	Pass	~10 hour long battery life
D-2	March 13, 2025	Jeremy Mark Tubongbanua	Pass	~6 hour long battery life
D-3	March 13, 2025	Emily Lai	Pass	~12 hour long battery life

Issues and Defects

Defect#	Action Item(s)	Improvement(s)
1	During test A-1, the camera was overheating.	Improvement was implemented such that camera inference can be controlled on demand. Continuous camera inference was causing the camera to overheat, so this method was opted out of.
2	Improper Camera Inference was common during test A-1.	Increasing the confidence threshold to 80% caused improper inferences to be less frequent.
3	Objects weren't being detected during test A-1.	Since the threshold was too high, the confidence threshold of the machine learning model was reduced to 60%.
4	Vibration Motors burnt out during test D-5	Due to an electrical issue, the vibration motors were burnt out and had to be replaced. To ensure this did not happen again, we ensured wires avoided touching each other using electrical tape.

Budget

Total budget: $1,062.12

Capstone Grou 6 Capstone Project Topic: Design and Development of a Blind Person Assistant

NOTE: ONLY out of pocket expenses directly relating to your Capstone project are eligible for reimbursement of $200 per group member for the 2024/25 Academic Year

Full Name of Capstone Student Group Members			
1	Andy Lau	6	
2	Emily Lai	7	
3	Jeremy Mark Tubongbanuaa	8	
4	Natasha Naorem	9	
5		10	

Date of Expens	Vendor and Description of Expense (ie. Bestbuy-HP Scanner)	Subtotal	HST	Total
10/8/2024	YISAVE TECHNOLOGY (HK) LIMITED - Lexar 32GC Micro SD Card 2 Pack	$18.99	$2.47	$21.46
10/8/2024	Shenzhen Shi WeiTu Keiji Youxian Gongsi - Raspberry Pi 5	$136.39	$17.73	$154.12
10/8/2024	Shen Zhen Shi Yi Chao Fan Wang Luo Ke Ji You Xian Gong Si - Aluminum Folding Cane	$25.50	$3.32	$28.82
2/27/2025	Shenzhenyingfengdamaoyiyouxiangongsi - Multicolored Dupont Wire 120pcs	$12.99	$1.69	$14.68
1/30/2025	Shenzhen Baihang Technology Co., Ltd - VEEKTOMX Mini Portable Charger	$32.98	$3.64	$36.62
9/17/2022	Otpos Technologies Inc - Raspberry Pi 4 Model B	$189.99	$0.00	$189.99
1/29/2025	ShenZhen HongYi New Tech. Investment Co.LTD - Gikfun Tactile Push Button	$13.88	$1.80	$15.68
9/17/2024	Luxonis Holding Corporation - Luxoniz Oak-D Lite Robotics Camera	$210.52	$0.00	$210.52
11/6/2024	ShenZhenShi JinBo MaoYi YouXianGongSi - WSZJINB Hook and Loop Strap, Cables	$15.99	$2.08	$18.07
11/1/2024	shenzhenyuandaokejiyouxiangongsi - OneLeaf Fastening Cable Ties	$12.57	$1.63	$14.20
11/1/2024	GUANGZHOU SENNUO TRADING COMPANY LIMITED - MEARCOOH Black Foam Roll	$18.99	$2.47	$21.46
10/22/2024	SHENZHEN LANG SI HU DONG KE JI YOU XIAN GONG SI	$17.59	$2.29	$19.88
9/18/2024	Xia ever nail supply inc - HONKYOB USB Mini Speaker	$17.99	$2.34	$20.33
9/7/2024	xingningshichuantianshiyeyouxiangongsi - tatoko Mini Virbation Motors 20pcs	$23.99	$0.00	$23.99
8/9/2024	Polymaker LLC - Polymaker PLA PRO Filament Light Blue	$31.99	$4.16	$36.15
8/9/2024	Polymaker LLC - Polymaker PLA PRO Filament Purple	$31.99	$4.16	$36.15
12/31/2024	Working Hours - Fall Semester	$100.00	$0.00	$100.00
3/31/2025	Working Hours - Winter Semester	$100.00		$100.00
				$0.00
	Max Eligible Amount: $800		Total Claim:	$1,062.12

Capstone Supervisor Approval - Signature

Student ID

Student Claimant - Print Name

Student Claimant - Signature

Dean (Dean's Delegate) Approval - Signature

I hereby certify that the above claimed expenses are intended for the sole purpose of the noted Capstone Project

Thank You

Any Questions?